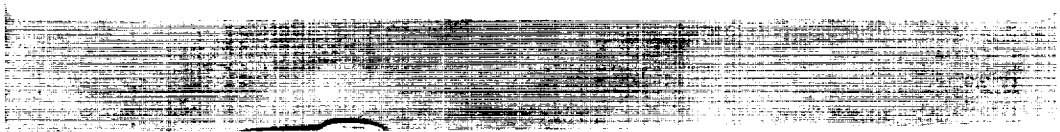
NASA Conference Publication 10111

Proceedings of the Third International Workshop on Neural Networks and Fuzzy Logic

Volume I

Proceedings of a workshop held at
Lyndon B. Johnson Space Center
Houston, Texas
June 1 - 3, 1992



NASA Conference Publication 10111

Proceedings of the Third International Workshop on Neural Networks and Fuzzy Logic

Volume I

Christopher J. Culbert, Editor
NASA Lyndon B. Johnson Space Center
Houston, Texas

Proceedings of a workshop held at
Lyndon B. Johnson Space Center
Houston, Texas
June 1 - 3, 1992

National Aeronautics and
Space Administration

January 1993

THIRD INTERNATIONAL WORKSHOP ON NEURAL NETWORKS AND FUZZY LOGIC

Program Schedule

Monday June 1, I992

7:30-8:00 Registration

8:00-8:30 Robert T. Savely, Chief Scientist, Information Systems Directorate, NASA/Lyndon B. Johnson Space Center, Houston, TX. **Welcoming Remarks.**

8:30-9:30 Jon Erickson, Chief Scientist, Automation and Robotics Division, NASA/Lyndon B. Johnson Space Center, Houston, TX. **Space Exploration Needs for Supervised Intelligent Systems.**

9:30-9:45 Break

Plenary Speakers

9:45-10:30 Piero P. Bonnisone, General Electric, **Fuzzy Logic Controllers: A Knowledge-Based Systems Perspective.**

10:30-11:15 Robert Farber, Los Alamos National Laboratory, **Efficiently Modeling Neural Networks on Massively Parallel Computers.**

11:15-1:00 Lunch

1:00-1:30 Lawrence O. Hall and Steve G. Romaniuk, University of South Florida, **Learning Fuzzy Information in a Hybrid Connectionist, Symbolic Model.**

1:30-2:00 Haluk Ogmen, University of Houston, **On the Neural Substrates Leading to the Emergence of Mental Operational Structures.**

2:00-2:30 Hao Ying, University of Texas Medical Branch, **A Fuzzy Controller with Nonlinear Control Rules is the Sum of a Global Nonlinear Controller and a Local Nonlinear PI-like Controller.**

2:30-2:45 Break

Parallel Sessions

2:45-3:15 Ron Maor and Yashvant Jani, Togai Infralogic, **Fuzzy Control of Electric Motors.**

Ronald Yager, Iona College, **A Hierarchical Structure for Representing and Learning Fuzzy Rules.**

3:15-3:45 Andre de Korvin and Margaret F. Shipley, University of Houston–Downtown, **Certain and Possible Rules for Decision Making using Rough Set Theory Extended to Fuzzy Sets.**

Jun Zhou and G. V. S. Raju, The University of Texas at San Antonio, **On Structuring the Rules of a Fuzzy Controller.**

4:15-4:45 Yashvant Jani, Togai Infralogic, Inc., Houston, TX, Gilberto Sousa, University of Tennessee, Knoxville, TN, Wayne Turner, Research Triangle Institute, Research Triangle Park, NC, Ron Spiegel and Jeff Chappell, U.S. EPA, Research Triangle Park, NC, **Fuzzy Efficiency Optimization of AC Induction Motors.**

Robert N. Pap, Mark Atkins, Chadwick Cox, Charles Glover, Ralph Kissel, and Richard Saeks, Accurate Automation Corporation, George C. Marshall Space Flight Center, **Advanced Telerobotic Control Using Neural Networks.**

6:00-7:00 **Wine and Cheese Reception**

7:00-9:00 **Banquet and Keynote Speaker**

Professor Bernard Widrow
Stanford University
Neural Controls

Tuesday, June 2, 1992

Plenary Speakers

8:00-8:45 Tomhiro Takagi, Laboratory for International Fuzzy Engineering Research, **Multilayered Reasoning Based by Means of Conceptual Fuzzy Sets.**

8:45-9:30 Michio Sugeno, Tokyo Institute of Technology, **Fuzzy Control of an Unmanned Helicopter.**

9:30-9:45 Break

9:45-10:15 Porter Sherman, Sikorsky Aircraft, **Fuzzy Logic Mode Switching in Helicopters**

10:15-10:45 James M. Urnes, Stephen E. Hoy, and Robert N. Ladage, McDonnell Aircraft Company, **A Neural Based Intelligent Flight Control System for the NASA F-15 Flight Research Aircraft.**

10:45-11:15 Capt. Gregory Walker, U.S. Army – NASA Langley Research Center, **A Teleoperated Unmanned Rotorcraft Flight Test Technique.**

11:15-11:45 James Villarreal, Robert N. Lea, Yashvant Jani, and Charles Copeland, NASA Lyndon B. Johnson Space Center, **Space Time Neural Network for Tether Operations in Space.**

11:45-1:00 **Lunch**

1:00-1:30 Hamid Berenji, Ames Research Center, **Structure Identification in Fuzzy Inference Using Reinforcement Learning.**

1:30-2:00 J. J. Buckley, University of Alabama at Birmingham, **Approximation Paper: Part 1.**

2:30-3:00 H. VanLangingham, A. Tsoukkas, V. Kreinovich, and C. Quintana, Virginia Polytechnic Institute and State University, University of Texas at El Paso, **Nonlinear Rescaling of Control Values Simplifies Fuzzy Control.**

3:00-3:15 **Break**

3:15-3:45 Robert Elliot Smith, The University of Alabama, **Genetic Learning in Rule-based and Neural Systems.**

3:45-4:15 Manuel Valenzuela-Rendon, Instituto Tecnologico de Estudios Superiores de Monterrey, **Evolving Fuzzy Rules in a Learning Classifier System.**

4:15-4:45 Charles L. Karr, U.S. Department of the Interior – Bureau of Mines, **Adaptive Process Control Using Fuzzy Logic and Genetic Algorithms.**

4:45-5:15 Hideyuki Takagi, University of California at Berkeley, **Design of Fuzzy Systems by Neural Networks and Realization of Adaptability.**

5:15-5:45 Paul P. Wang, Duke University, **Improvement of Fuzzy Controller Design Techniques.**

Wednesday, June 3, 1992

Plenary Speakers

8:00-8:45 Jim Bezdek, University of West Florida, **Two Generalizations of Kohonen Clustering.**

8:00-8:45 Jim Keller, University of Missouri, **Possibilistic Clustering for Shape Description.**

9:30-9:45 Break

Parallel Sessions

9:45-10:15	Todd Espy, Endre Vombrack, and Jack Aldridge, Togai Infralogic, **Application of Genetic Algorithms to Tuning Fuzzy Control Systems.**	Kaoru Hirota, Hosei University – Tokyo, **3D Image Recognition Based on Fuzzy Neural Network Technology.**
10:15-10:45	Isao Hayashi, Elichi Naito, Jun Ozawa, and Noboru Wakami, Matsushita Electric Industrial Co., Ltd., **A Proposal of Fuzzy Connective With Learing Function and its Applications to Fuzzy Retrieval Systems.**	Chul-Sung Kim, Exxon Production Research, **Application of Fuzzy Set and Dempster-Shafer Theory to Organic Geochemistry Interpretation.**
10:45-11:15	Sujeet Shenoi, Chun-Hsin Chen, and Arthur Ramer, University of Tulsa, University of New South Wales, **Towards Autonomous Fuzzy Control.**	Ben Jansen, University of Houston, **Determining the Number of Hidden Units in Multi-layer Perceptrons using F-Ratios.**
11:15-11:45	P.A. Ramamoorthy and Shi Zhang, University of Cincinnati, **A New Approach for Designing Self-Organi zing Systems and Applications to Adaptive Control**	Sunanda Mitra, Texas Tech University, **Adaptive Fuzzy System for 3D Vision.**

11:45-1:00	**Lunch**
1:00-1:30	Michael Murphy, University of Houston – Downtown, **Fuzzy Logic Path Planning System for Collision Avoidance by an Autonomous Rover Vehicle.**
1:30-2:00	Francois G. Pin and Yutaka Watanabe, Oak Ridge National Laboratory, **Driving a Car with Custom-Designed Fuzzy Inferencing VLSI Chips and Boards.**
2:00-2:30	Enrique Ruspini, Stanford Research Institute, **Autonomous Vehicle Motion Control, Approximate Maps, and Fuzzy Logic.**
2:30-2:45	**Break**
2:45-3:15	John Yen and Nathan Pfluger, Texas A&M University, **A Fuzzy Logic Controller for an Autonomous Mobile Robot.**
3:15-3:45	Sandeep Gulati and Raoul Tawel, Jet Propulsion Laboratory, **Intelligent Neuroprocessors for In Situ Propulsion Systems Health Management.**
3:45-4:15	Reza Langari, Texas A&M University, **Synthesis of Non-Linear Control Strategies from Fuzzy Logic Control Algorithms.**
4:15-4:45	P.Z. Wang, Hongmin Zhang, and Wei Xu, National University of Singapore, **Truth-Valued-Flow Inference (TVFI) and Its Applications in Approximate Reasoning.**

CONTENTS

Volume I

Fuzzy Logic Controllers: A Knowledge–Based System Perspective

Piero P. Bonissone

Artificial Intelligence Laboratory
GE Corporate Research and Development
Bldg. K1-5C32A, PO Box 8
Schenectady, NY 12301
Bonissone@crd.ge.com

Over the last few years we have seen an increasing number of applications of Fuzzy Logic Controllers. These applications range from the development of auto-focus cameras, to the control of subway trains, cranes, automobile sub-systems (automatic transmissions), domestic appliances, and various consumer electronic products.

A Fuzzy Logic Controller is a knowledge based system in which the knowledge of process operators or product engineers has been used to synthesize a closed loop controller for the process. We will compare the development and deployment of Fuzzy Logic Controllers (FLC) with that of Knowledge Based System (KBS) applications.

Traditional controllers are derived from a mathematical model of the open-loop process to be controlled, following classical control theory techniques. FLCs are typically derived from a knowledge acquisition process (or are automatically synthesized from a self-organizing control architecture). In either case, the result of the synthesis is a Knowledge Base (KB), rather than an algorithm. The KB consists of a set of fuzzy-rules (rules and termsets), which is evaluated by an interpreter. The interpreter is composed of a quantification (or fuzzification) stage, an inference engine (or fuzzy matcher), and a defuzzification stage.

We will analyze FLCs according to three organizing layers typically used in describing Knowledge Based Systems: knowledge representation, inference, and control. In the knowledge representation layer we will describe fuzzy state vectors, term-set of linguistic values, and fuzzy production rules. In the inference layer we will provide a geometric interpretation (for the disjunctive case) of the generalized modus ponens, and describe the inference process based on fuzzy predicate evaluation, rule Left Hand Side (LHS) evaluation, rule detachment, and rules aggregation. In the control layer we will show three different defuzzification methods and we illustrate meta-reasoning capabilities (supervisory mode).

FLC interpreters are used during the development phase of a FLC application to provide inference traceability (transparency), which facilitates the KB design, implementation, and refinement. However, the use of an interpreter requires the evaluation of all the rules in the KB at every iteration.

Therefore, after a functional validation (stability or robustness analysis), the KB is compiled like a programming language or a traditional knowledge base application, and a simpler run-time engine is used for deployment. The result of this compilation process is a look-up table that allows for a faster, more efficient execution that can be performed by simpler processors. Not only is the response time reduced, but the memory requirements are so drastically decreased that it is possible to implement the FLC using very small amounts of memory. This feature enables us to build inexpensive FLCs for cost-sensitive applications.

In summary, we consider a Fuzzy Logic Controller to be a high level language with its local semantics, interpreter, and compiler, which enables us to quickly synthesize non-linear controllers for dynamic systems.

Efficiently Modeling Neural Networks on Massively Parallel Computers

Robert M. Farber
Los Alamos National Laboratory
Los Alamos, N.M.
87544

Neural networks are a very useful tool for analyzing and modeling complex real world systems. Applying neural network simulations to real world problems generally involves large amounts of data and massive amounts of computation. To efficiently handle the computational requirements of large problems, we have implemented at Los Alamos a highly efficient neural network compiler for serial computers, vector computers, vector parallel computers, and fine grain SIMD computers such as the CM-2 connection machine. This paper will describe the mapping used by the compiler to implement feed-forward backpropagation neural networks (D. Rummelhart and J. McClelland 1986) for a SIMD (Single Instruction Multiple Data) architecture parallel computer. Thinking Machines Corporation has benchmarked our code at 1.3 billion interconnects per second (approximately 3 gigaflops) on a 64,000 processor CM-2 connection machine (Singer 1990). This mapping is applicable to other SIMD computers and can be implemented on MIMD computers such as the CM-5 connection machine. Our mapping has virtually no communications overhead with the exception of the communications required for a global summation across the processors (which has a sub-linear runtime growth on the order of O(log(number of processors))). We can efficiently model very large neural networks which have many neurons and interconnects and our mapping can be extend to arbitrarily large networks (within memory limitations) by merging the memory space of separate processors with fast adjacent processor inter-processor communications. This paper will consider the simulation of only feed forward neural network although this method is extendible to recurrent networks.

A simple XOR network can be seen in Fig 1. This network (or any feed-forward neural network) is "trained" as follows: First, the outputs for each example of a "training set" of examples are calculated for a given set of network parameters (neuron thresholds and connection weights). This can be seen for the XOR problem of fig 1 in eqn. 1.1 - 1.4. In these equations W(a,b) means the connection weight from a to b and g() is a user specified linear or non-linear function. The fitness of the calculated outputs (and

$$H = H_{threshold} + W(I_1,H) * I_1 + W(I_2,H) * I_2 \qquad \text{Eqn 1.1}$$

$$O = O_{threshold} + W(I_1,O) * I_1 + W(I_2,O) * I_2 \qquad \text{Eqn 1.2}$$

$$O \mathrel{+}= g(H) * W(H,O) \qquad \text{Eqn 1.3}$$

$$O = g(O) \qquad \text{Eqn 1.4}$$

hence the network parameters) is determined by some function of the known and calculated outputs. A common fitness function is the sum of the square of the differences as shown in eqn. 2. The parameters of the network are then adjusted by some nonlinear

$$\text{Fitness} = \sum_{0}^{num\_examples} (known\_output - calculated\_output)^2 \qquad \text{Eqn 2}$$

minimization scheme such as powell's method or conjugant gradient (Press *et. al.*_1988). The network is continually adjusted and re-evaluated until a "best fit" is found. The neural network is then said to be "trained". If the number of examples is small relative to the

number of network parameters, then the network can "memorize" the training set. In other words, there are so many parameters in the network that it "memorizes" the training set. Unfortunately, neural networks which are over parameterized generally predict poorly on examples which were not in the training set. Hence most neural networks are trained with a number of examples far larger than the number of network parameters. This "overloading" of the network is done to force the network to "generalize" a solution from the training set. It is then hoped that the network will then predict well on data which was not in the training set. The literature abounds with important problems where neural networks have been shown to be good predictors. For example, neural networks can be used to predict time series with orders of magnitude increases in accuracy over conventional methods (Lapedes and Farber 1987 and Lapedes and Farber 1987). Neural networks have also been shown to be highly accurate predictors of coding regions for short regions of DNA (Farber *et. al.* 1992 and Lapedes and Farber 1989). We can see that the runtime growth for evaluating a neural network during training is on the order of $O(m*n)$ where m is the number of network parameters and n is the number of examples. From our discussion we can see that n generally dominates the runtime growth.

This means that contrary to what one would first expect, the most efficient method of mapping neural networks on to a massively parallel machine is not one neuron per processor. Rather, the most efficient method is to map *one example to each processor*. By using this mapping for SIMD or MIMD (Multiple Instruction Multiple Data) parallel computers, it is possible to get number_of_example operations done in each instruction cycle of the machine by having each processor evaluate the network for it's example. Hence, we effectively get no change in our runtime for a problem which has one example over a problem which has 250,000 examples. In reality, there will be a small increase in the runtime as the number of examples exceeds the number of processors. However this increase is on the order O(number_of_examples/number_of_processors) and is very small for the large numbers of processors in current SIMD machines. Thus, we get essentially large training sets for free. This allows neural networks to be applied to problems of a size and complexity not possible using serial machines. Our mapping can also be used to efficiently implement neural networks on vector computers. However, the runtime growth is much more strongly affected by the number of examples (effectively, the number of processors is small). Thus conventional vector machines such as a CRAY cannot achieve the reduction in the runtime growth possible with a SIMD machine containing a large number of processors. This analysis is overly simplistic since there are complex trade-offs between cycle time, vector pipeline length, and the number of processors. The bottom line is that given access to both vector machines and highly parallel SIMD/MIMD machines, we use vector machines for medium sized problems (generally less than 8,000 examples) and parallel machines for larger problems (from 8,000 examples to 10^6 examples).

The overall computational efficiency of a parallel computer can be high only as long as the associated communications overhead for the problem is low. Otherwise the parallel processors will spend all their time waiting for data. Using our mapping onto SIMD hardware, we will show that it is possible to avoid any communications overhead by mapping neural networks onto the parallel machine via the one example per processor approach. In our implementation on the CM-2 connection machine, the only communications required (with one minor exception) are global broadcast and local processor to processor communications. Since both of these operations occur in one clock cycle on the

connection machine, they provide no delay over a simple memory fetch. Hence the rate limiting step is how fast the parallel hardware of the connection machine can do floating point operations. In other words, our mapping turns the training of neural networks into a parallel algorithm which is limited by the computational rate of the hardware and not by communications overhead.

The mapping onto the CM-2 for the XOR architecture of fig 1 can be seen in fig 2-4. As can be seen in fig 2, the front-end computer contains all the network parameters and the SIMD processors contain all examples and temporary storage for the network. We can see the initial calculation of the hidden neuron (given in eqn 1.1) as it would be executed in parallel in Fig 3 - 4. The feedforward pass is initiated by broadcasting the neuron threshold from the front-end computer to all processors (see fig 3). The connection weight $W(I_1,H)$ is then broadcast to all processors with the instruction to multiply it by the local memory location containing the value of I_1 and add it to the local memory location containing the value of the hidden neuron (see fig 4). Since each SIMD processor contains one example, we get the number_of_examples instructions done per instruction cycle with no communications overhead. Similarly the calculations of eqn 1.2 - 1.4 occur using only global broadcast and local processor memory. It is clear that we are able to calculate the outputs for all the training examples for the XOR architecture or any arbitrary neural network, without communications delays, using only global broadcast communications. (The evaluation of recurrent networks is dependent upon how the back connections are to be evaluated. It is possible to do a purely parallel implementation for SIMD architectures using our mapping (see Pineda 1988 for the mathematical description). Other recurrent implementations may require a MIMD architecture as the required number of conditional operations would result in an extremely inefficient use of the SIMD processors per machine cycle.) The next step is to evaluate how the calculated outputs fit the known outputs. To do this the front-end issues an instruction to subtract the known output from the calculated output and square the result. Since all memory values are in local processor memory there is no communications overhead. The front-end then issues an instruction to calculate the summation over all processors of the squared differences. On the CM-2, the global summation instruction is provided by Thinking Machines Corporation and is optimized for their hardware. However, the global summation instruction has a runtime growth which is approximately O(log(n)) where n is the number of processors. Fig 5 diagrams how a O(log(n)) runtime growth could be achieved for a global summation. Since the run-time growth of this instruction is sub-linear with respect to the number of processors (or number of examples for our problem), it does not provide a significant decrease in the runtime performance. All other network calculations required for backpropagation occur in a similar manner and have no communications overhead except for that required by the global summation over processors.

Our mapping of one example per processor also allows networks with large numbers of parameters to be trained. We can see in Fig 6 that the worst-case memory growth for a fully interconnected recursive neural network is on the order $O(n^2)$; where n is the number of neurons. Since the network parameters (neuron thresholds and connections weights) are the same for all examples and hence for all the SIMD processors, it is makes sense to store them in one common block of memory and broadcast them to all other processors. This makes for an ideal mapping onto the CM-2 hardware as the $O(n^2)$ network parameters can be stored in the large virtual memory space of the front-end computer and broadcast to the SIMD processors. This frees the limited memo-

ry available to each CM processor to be used for the storage of the example input(s) and output(s) and intermediate values of the calculations.

It is the memory available to each SIMD processor which limits the size of the neural network, the size of individual training examples, and the amount of training data which can be evaluated. If the SIMD hardware has fast adjacent processor communications it is possible to efficiently merge the memory of adjacent processors to allow arbitrarily large training examples and neural networks to be evaluated. (It is possible to use a fast I/O memory device for the SIMD processors such as the CM-2 data vault to allow essentially unlimited network sizes and number of examples. However, we have not found it necessary to go to such extremes to train complex networks with even 10^5 to 10^6 examples.) This means that the memory map of individual SIMD processors will differ. However, we can merge the memory space of different processors by defining a special memory location in the memory map of all the SIMD processors to be a memory data bus. If we consider the example of fig 4 in evaluating an example of the XOR network, we would see a mapping onto the SIMD processor as seen in fig 7. If a value is required in the first processor which is in the memory of the second processor, it is copied to the common memory bus location and transferred via adjacent processor communications to the first processor. The arithmetic operation then proceeds on the first processor. Data shifts between adjacent processors on the CM-2 connection machine occur in one machine cycle (which is as much as 10^3 time faster than using the router communications). Thus we incur minimal communications overhead when using merged processors. However, merging processors introduces inefficiencies other than in moving data between the memory space of separate processors. In the case of merging two processors, only half of the SIMD processors can be active per computational instruction cycle. Similarly only 1/3 of the processors would be active if three processors were merged together and so forth. The advantage of the merged memory model is that arbitrarily large neural networks and data sets (which normally would be impossible to evaluate due to memory limitations) can be evaluated and in a manner transparent to the user. Since the number of processors which have to be merged to provide adequate memory storage is quite small in most cases, the performance loss is quite acceptable.

At Los Alamos, we have been using the mappings described above within the context of a neural network compiler since 1988. The details of the compiler are too numerous to present here. However, the compiler implements a paradigm familiar to any code developer as seen in fig 8. Aside from receiving the problem specification (the neural network architecture, initial parameter values, and training data) the compiler does all the remaining steps automatically for the destination machine including "writing" of the neural network program. Fig 9 shows how data moves through a complete neural network simulation. We can see that the neural network can be specified interactively by a graphical interface or by a machine generated file. The graphical interface allows a user to merge sub-networks "trained" to task into a large complex network. The sub-networks parameters may be locked to preserve the functionality of the sub-network or they may be "equivalenced" to force the unique sub-network parameters to maintain identical values during training. Of course the user may "unlock" the network parameters at will to allow "tweaking" of the parameter for the particular problem. The user may also automatically generate the network architecture so that the neural network may be modified so that various "pruning" or "growing" heuristics may be used. The training set data is presented to the compiler as either floating point or single bit boolean values. This allows

the compiler to minimize floating point operations for the individual training set and can provide significant increases in computational throughput. The data manipulation prior to the compiler can be a non-trivial task. We have had intermediate amounts of data exceeding 60 gigabytes which had to be pre-processed prior to presentation to the compiler.

The compiler takes the network/data specification and generates an intermediate language "program". This program then goes through a dependency analysis and is presented to a "compiler" which creates a relocatable instruction stream which is then passed to the loader linker. For the CM-2 the loader/linker creates an appropriate memory map (including merging multiple processors together) and creates a state machine instruction stream which is then executed once the data is loaded into the connection machine.

The compiler automatically calls the user specified optimization code as well as user functions specifying arbitrary neuron types. The user code on the front-end computer sees the compiler generated calculation of the forward pass, error propagation and calculation of the gradient (if possible) for the destination machine given the specified training set and neural network architecture. The user can then call these routines from their optimization code. Since algorithms for nonlinear or multidimensional optimization are quite complex and are either difficult or impossible to implement efficiently on a SIMD processor array, they are instead executed serially on the front-end computer. This allows the use of optimization algorithms such as conjugant gradient, powell's method, steepest descents or some other algorithm written in the users favorite language. This use of the front-end provides advantages on the connection machine. For example, the optimization code can "twiddle" network parameters with a 100 ns clock instead of the μs cycle time of the connection machine processors. In addition, some of the work done in the optimization code can be gotten "for free" due to the asynchronous operation of the front-end computer and the SIMD array of processors.

In summary, we have been able to exploit the gigaflop capabilities of the connection machine to train arbitrary feed forward neural networks on large, complex, and noisy data sets with examples on the order of hundreds of thousands to millions. We have done this with a neural network compiler which implements an extremely efficient mapping to SIMD architecture parallel computers. The mapping allows efficient use of the computational facilities of the parallel hardware with virtually no communications overhead. Arbitrarily large networks can be implemented by using the large virtual address space of the front end computer and by merging the memory space of adjacent SIMD processors together via fast local inter-processor communications. Thinking Machines Corporation has acknowledged that our implementation is considerably faster than other known implementations and that our implementation "has either constant time behavior or linear time dependence with respect to the number of training patterns, depending on the size of the connection machine used" (Singer 1990). Since the number of examples is the dominating factor in the runtime growth of training, our method allows the use of the CM-2 Connection Machine for real world problems of a complexity not possible using other computational hardware.

This work was done under the auspices of the U. S. Department of Energy and was partially funded by a grant from the National Institutes of Health (GM 40789-03). We express our gratitude for the hospitality of the Santa Fe Institute where part of the work was performed. We also acknowledge the help and support of Alan Lapedes who has been an integral part of the design and use of this work and without whom this work would have been impossible.

Fig 1: An XOR Neural Network

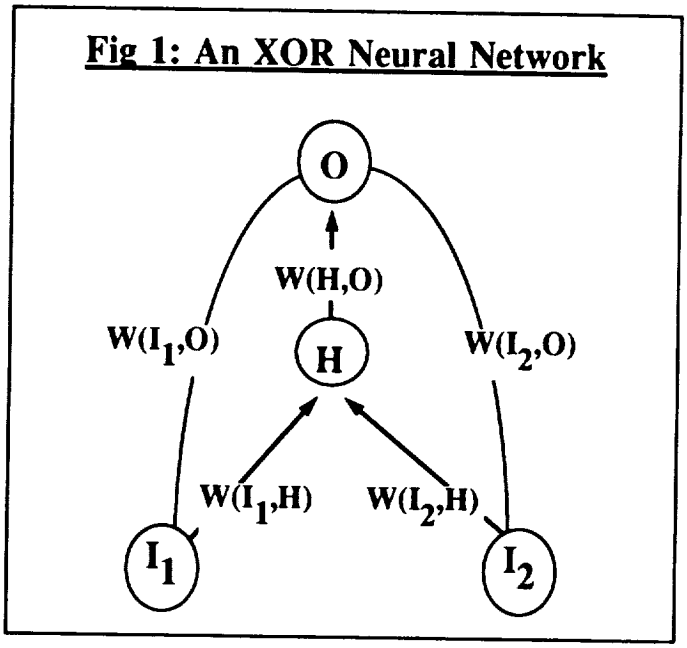


Fig 2: Mapping onto the Connection Machine

The Front End Computer (Generally a SUN) holds all variables for the neural network in *virtual memory*. This allows essentially unlimited neural network sizes and connectivity. The Front End also contains the energy minimization code written in a high level language like C. We generally use conjugant gradient although the user has complete flexibility to use his own code.

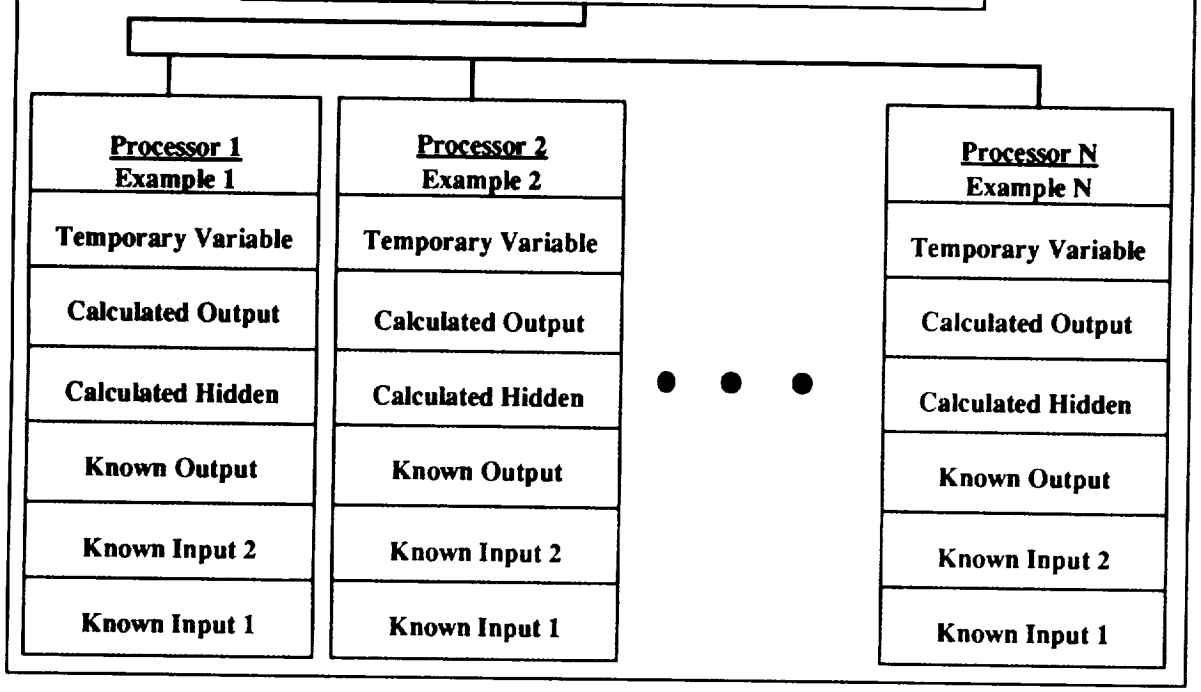


Fig 3: Example of a global broadcast

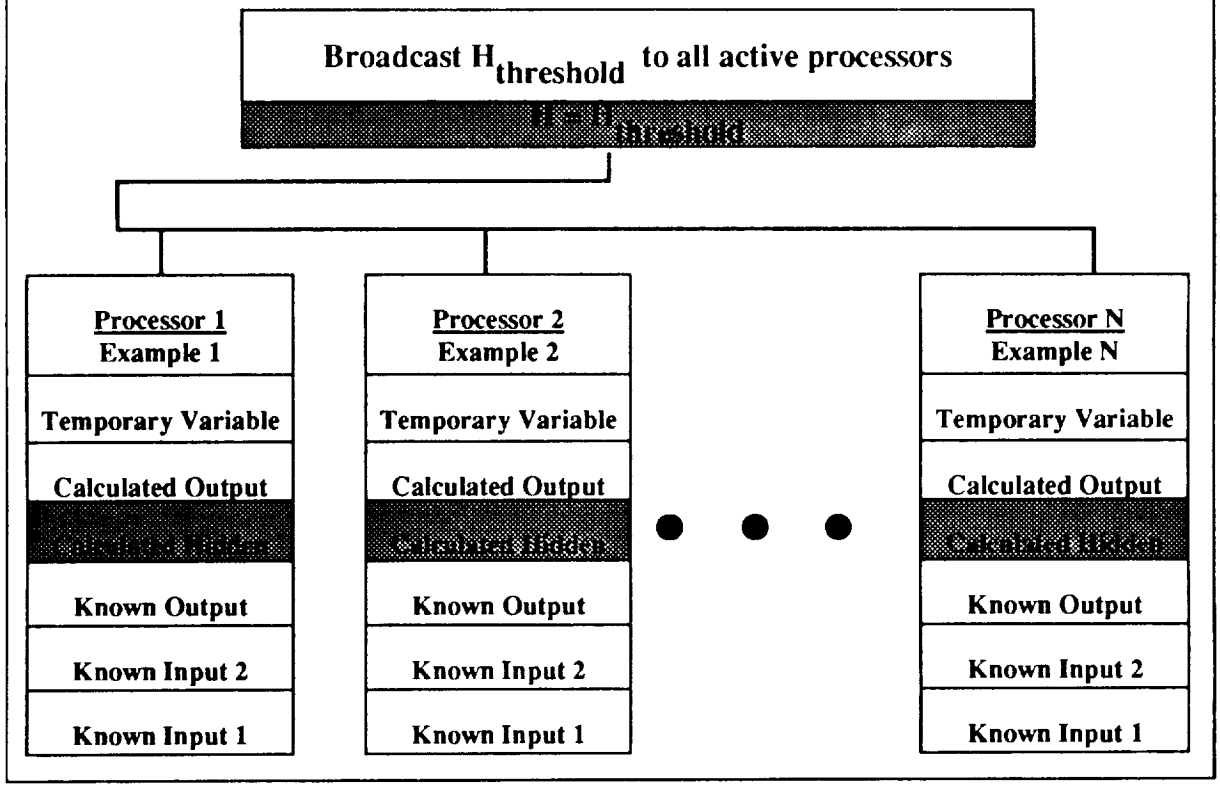


Fig 4: Calculation of Hidden Neuron for XOR Network. All operations are local except for the global broadcast of connections $W(I_1,H)$ and $W(I_2,H)$.

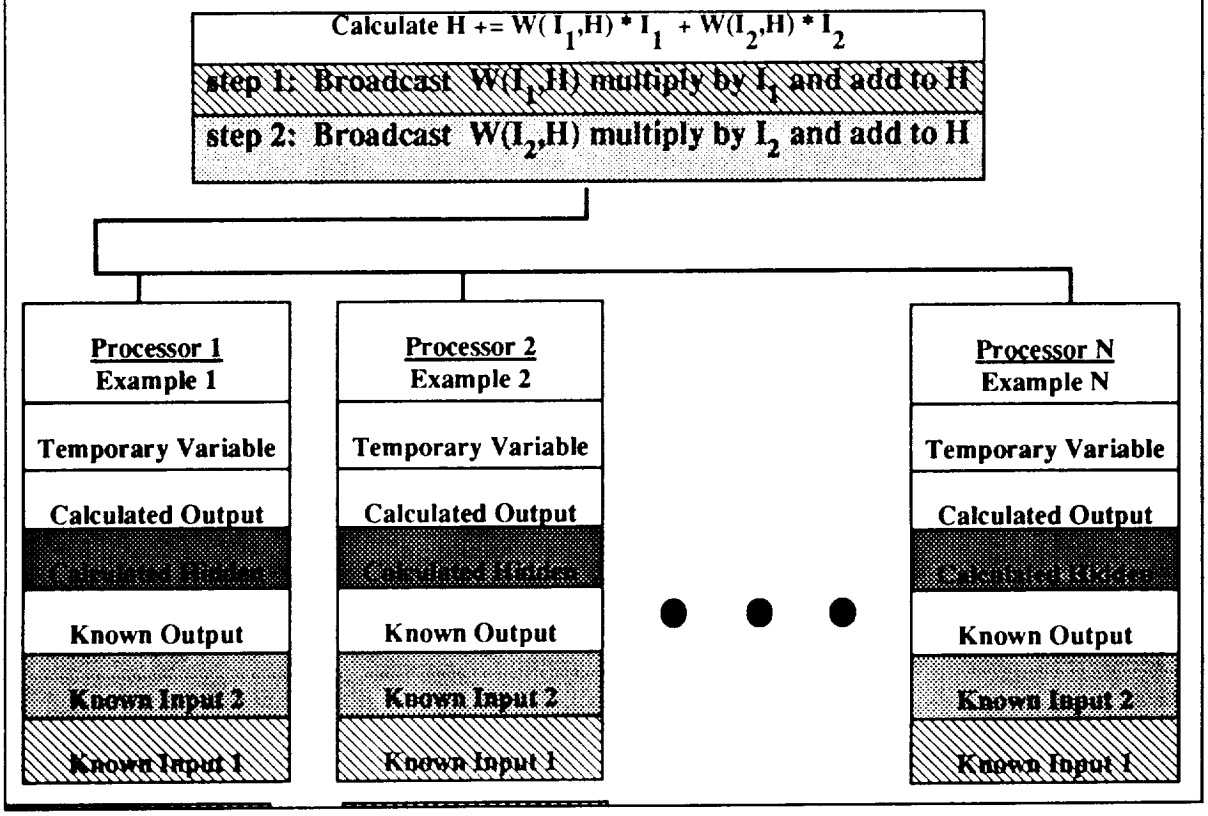


Fig 5: log(n) Runtime Growth of Global Summation

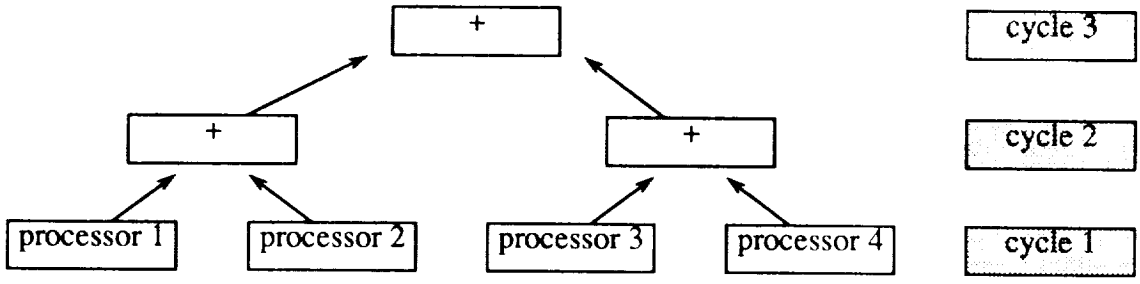
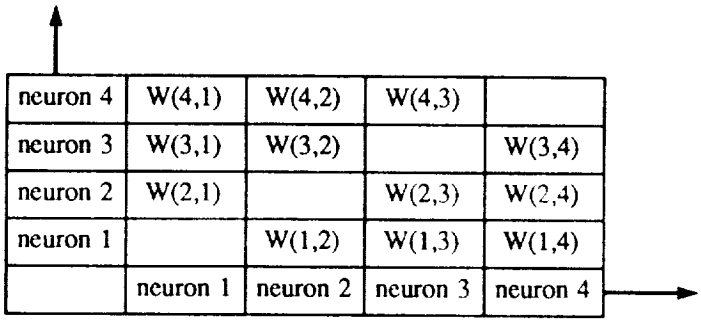


Fig 6: $O(n^2)$ Memory Growth for a Fully Interconnected Neural Network



neuron 4	W(4,1)	W(4,2)	W(4,3)	
neuron 3	W(3,1)	W(3,2)		W(3,4)
neuron 2	W(2,1)		W(2,3)	W(2,4)
neuron 1		W(1,2)	W(1,3)	W(1,4)
	neuron 1	neuron 2	neuron 3	neuron 4

Fig 7: Calculation of Hidden Neuron (eqn 1.2) for XOR Network. All operations are local except for the global broadcast of the connections $W(I_1,H)$ and $W(I_2,H)$ and local inter-processor communications.

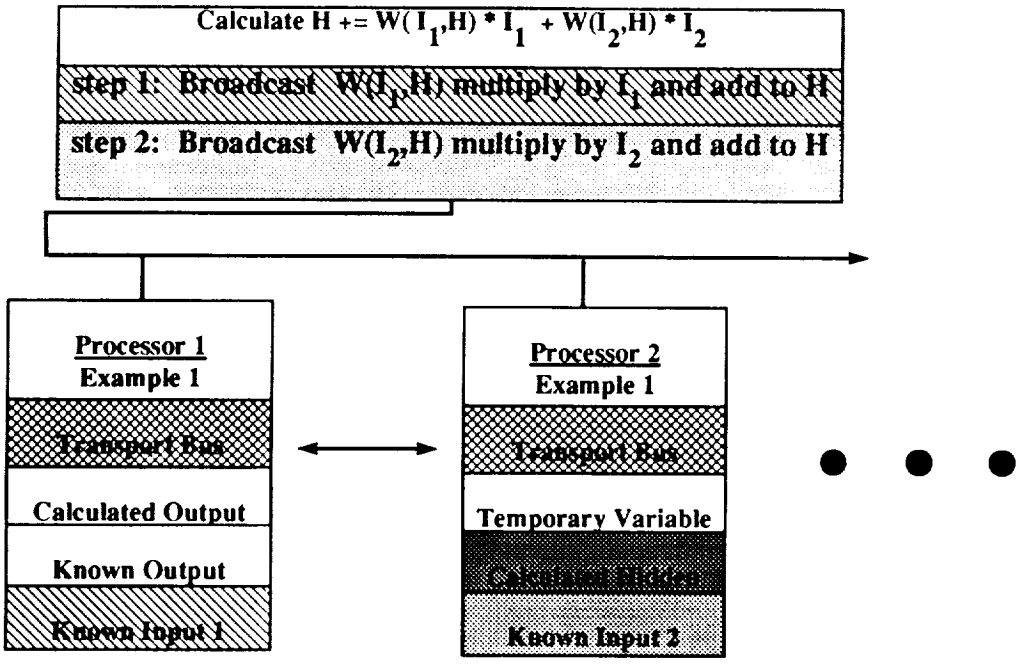


Fig 8: Compiler Paradigm

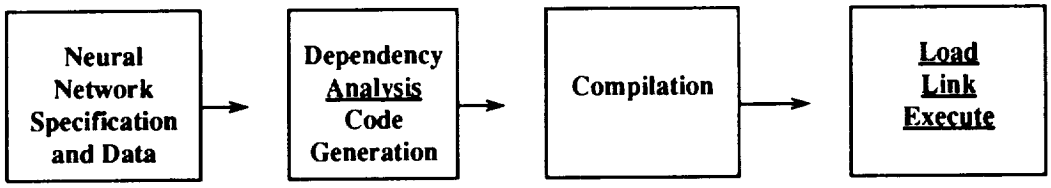
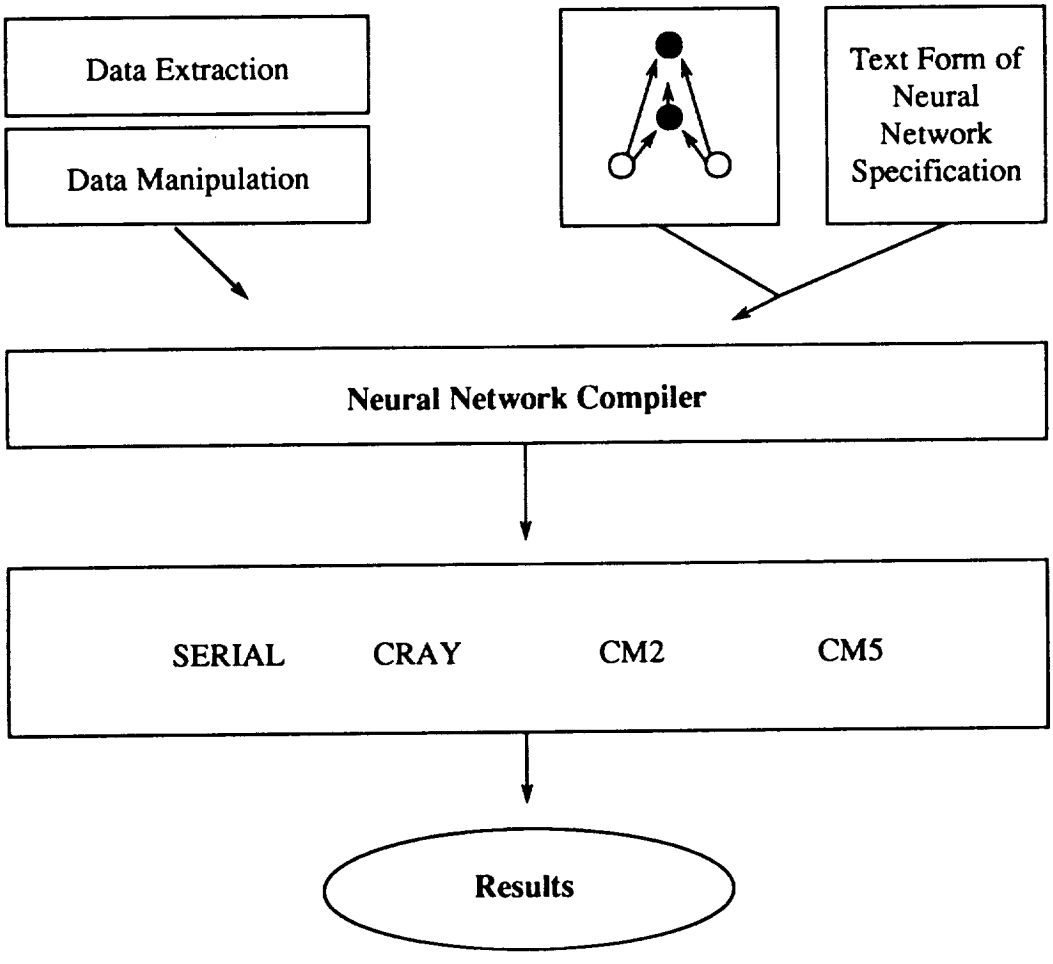


Fig 9: Block Diagram of Neural Network System



References

A. S. Lapedes, R. M. Farber, "*How Neural Networks Work*" Proceedings of the IEEE, Denver Conference on Neural Networks, 1987.

A. S. Lapedes, R. M. Farber, *"Non-linear Signal Processing using Neural Networks, Prediction and System Modeling "*, LANL technical report LA-UR-87-2662

A. S. Lapedes, C. Barnes, C. Burks, R. M. Farber, K. Sirotkin, *"Application of Neural Networks and Other Machine Learning Algorithms to DNA Sequence Analysis"* published in Computers and DNA, SFI Studies in the Sciences of Complexity, vol. VII (1989).

R. M. Farber, A. S. Lapedes, K. Sirotkin, *"Determination of Eukaryotic Protein Coding Regions Using Neural Networks and Information Theory"*, J. Mol. Biol. (1992) **226**, 471-479.

F. J. Pineda, *"Generalization of Backpropagation to Recurrent and Higher Order Neural Networks"*, Neural Information Processing Systems, Dana Z. Anderson editor, pg. 602-611.

W. H. Press, B. P. Flannery, S. A. Teukolsky, W. T. Vetterling, *"Numerical Recipes in C"*, Cambridge University Press, 1988.

D. Rummelhart, J. McLelland, *"Parallel Distributed Processing"*, Vol. 1, M.I.T. Press, Cambridge, MA. (1986)

Singer A., *"Implementations of Artificial Neural Networks on the Connection Machine"*, Parallel Computing, **14**:305-315, 1990

Learning Fuzzy Information in a Hybrid Connectionist, Symbolic Model

Steve G. Romaniuk and Lawrence O. Hall

Department of Computer Science and Engineering

University of South Florida

Tampa, FL 33620

e-mail: romaniuk@daffy.csee.usf.edu, hall@waterfall.csee.usf.edu

Abstract

An instance-based learning system is presented. SC-net is a fuzzy hybrid connectionist, symbolic learning system. It remembers some examples and makes groups of examples into exemplars. All real-valued attributes are represented as fuzzy sets. The network representation and learning method is described. To illustrate this approach to learning in fuzzy domains, an example of segmenting magnetic resonance images of the brain is discussed. Clearly, the boundaries between human tissues are ill-defined or fuzzy. Example fuzzy rules for recognition are generated. Segmentations are presented that provide results that radiologists find useful.

1 Introduction

This paper describes the use of a hybrid connectionist, symbolic machine learning system, SC-net [4, 8], to learn rules which allow the discrimination of tissues in magnetic resonance (MR) images of the human brain. Specifically, a 5mm thick slice in one spatial orientation will be used to illustrate SC-net's capabilities. The problem involves identifying tissues of interest which include gray matter, white matter, cerebro-spinal fluid (csf), tumor when it exits, edema and/or necrosis. Essentially, a segmentation of the MR image into tissue regions is the aim of this research. The training data is chosen by a radiological technician who is also familiar with image processing and pattern recognition.

SC-net is an instance-based learning system. It encodes instances or modifications of instances in a connectionist architecture for use in classification after learning. Fuzzy sets are directly represented by groups of cells in the network. Membership functions for any defined fuzzy sets are also learned during the training process with the dynamic plateau

modification feature of SC-net [7].

The rest of this paper will consist of a description of the relevant features of the SC-net learning system, a description of the processing of a MR image slice, the presentation and discussion of the segmentation results obtained with the SC-net system, a discussion of how these results compare with other techniques that have been used [5] and an analysis of the feasibility of the SC-net approach in this domain.

2 The SC-net approach

Each cell in an SC-net network is either a min, max, negation or linear threshold cell. The cell activation formulae are shown in Figure 1. The output structure of the network is set up to collect positive and negative evidence for each output. For an output cell in a classificatory domain, an output of 0 indicates no presence, 0.5 indicates unknown and 1 indicates true. We will show an example of a different us of the output values in the MR image segmentation domain.

SC-net configures its connectionist architecture based upon the training examples presented to it. The learning algorithm responsible for the creation of the network topology is the Recruitment of Cells algorithm (RCA) [4, 7]. RCA is an incremental, instance-based algorithm that requires only a single pass through the training set. Every training instance is individually presented to the network for a single feedforward pass. After the pass has been completed, the actual and the expected activation for every output are compared. Three possible conditions may result from this comparison.

- The example was correctly identified (error is below some epsilon). No modifications are made to the network.
- The example is similar to at least one previously seen and stored instance (error within 5 epsilon). For those output cells that have an activation within 5 epsilon of the expected output, a bias is adjusted to incorporate the new instance.
- The example could not be identified by the network. This results in the recruitment of

CA_i - cell activation for cell C_i.
O_i - output for cell C_i in [0,1].
$O_{i_{positive}}$ and $O_{i_{negative}}$ are the positive and negative collector cells for C_i respectively.
$CW_{i,j}$ - weight for connection between cell C_i and C_j, $CW_{i,j}$ in R.
CB_i - cell bias for cell C_i, CB_i in [-1..+1].

$$CA_i = \begin{cases} min_{j=0,..,i-1,i+1,..n}(O_j * CW_{i,j}) * |CB_i| & C_i \text{ is a } min \text{ cell} \\ max_{j=0,..,i-1,i+1,..n}(O_j * CW_{i,j}) * |CB_i| & C_i \text{ is a } max \text{ cell} \\ \left|\sum_{j=0;j\neq i}^{n} O_j * CW_{i,j}\right| * CB_i & C_i \text{ is a } ltc \text{ cell} \\ 1 - (O_j * CW_{i,j}) & C_i \text{ is a negate cell} \\ O_{i_{positive}} + O_{i_{negative}} - 1/2 & C_i \text{ is either an intermediate} \\ & \text{or } final \text{ output cell.} \end{cases}$$

$$O_i = max(0, min(1, CA_i))$$

Figure 1: Cell activation formula

a new cell (referred to as an information collector cell, ICC). Appropriate connections from the network inputs to the ICC are created. The ICC cell itself is connected to either the positive (PC) or negative collector (NC) cell. The PC is used to collect positive evidence, whereas the NC accumulates negative evidence. The initial empty network structure for a two input (one output) fuzzy exclusive-or is presented in Figure 2. Note that the uk cell always takes an activation of 0.5. The complete learned network for the fuzzy exclusive-or is shown in Figure 3, where cells c1-c3, c5 are IC cells and n1, n2, c4, and c6 are negation cells.

To improve on the generalization capabilities of the RCA generated SC-net network a form of post training generalization is employed. This method is called the min-drop feature. Whenever a test pattern is presented to the system, which cannot be identified by any of the output cells, the min-drop feature is applied. If a new pattern cannot be recognized by the network, all output cells will be in an inactive state (an unknown response of 0.5 is returned). In this case the min-drop feature is applied to find the nearest corresponding

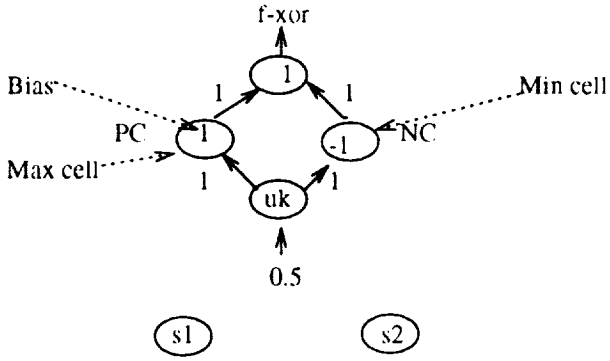


Figure 2: SC-net structure

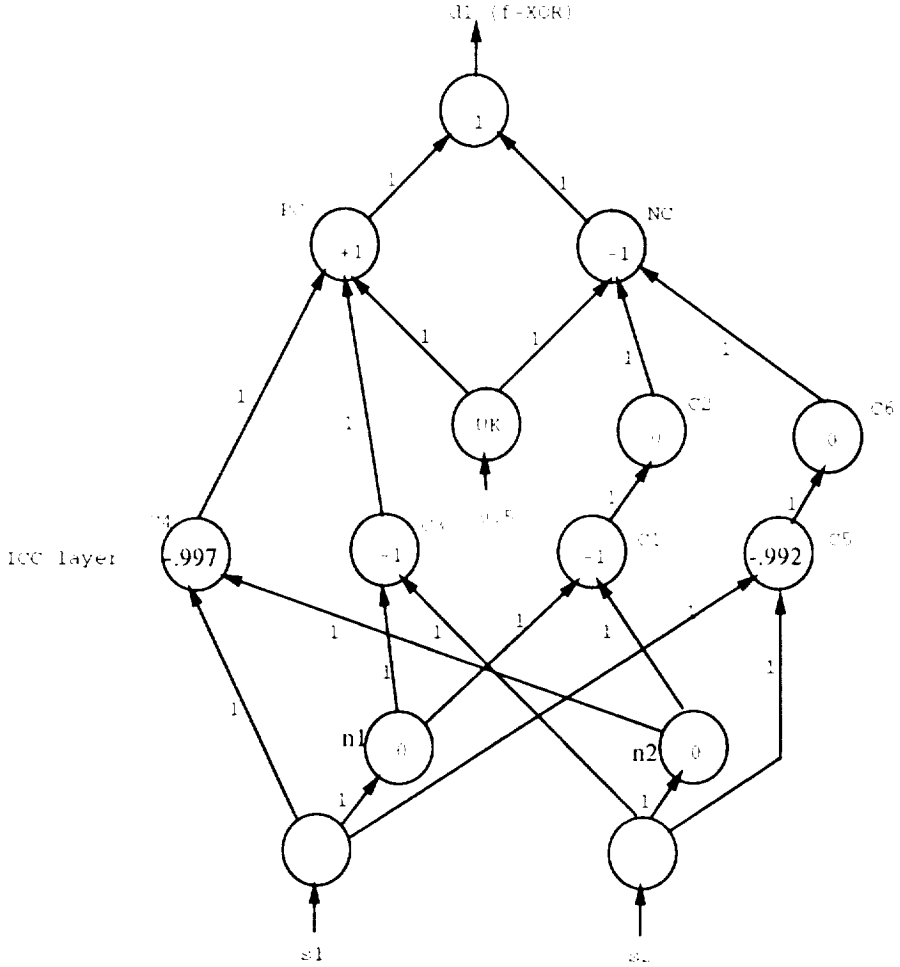


Figure 3: The network for the fuzzy exclusive-or

output for the current pattern. New patterns are stored in the network through recruitment of IC cells (and possibly some negation cells). These IC cells are essentially min-cells, which return the minimum of the product formed from the incoming activation and the weight on the corresponding connection. The min-drop feature works by dropping (ignoring) the next piece of evidence which is below some threshold. The process is repeated until one or more output cells enter an active state (fire). The final number of connections dropped indicates the degree of generalization required to match the newly presented pattern. In a second mode, a bound may be placed on the min-drop value, preventing an unwarranted over-generalization. RCA and post training generalization in the form of the min-drop feature provide good generalization. However, several problems can be associated with the RCA learning phase.

- Network growth can be linear in the number of training examples.
- As a direct consequence of the first problem storage and time (to perform a single feedforward pass) requirements may increase beyond the networks physical limitations.

- Generalization on yet unseen patterns is limited, and requires use of min-drop feature.

To address the above problems a network pruning algorithm was developed. The GAC (Global attribute Covering) algorithm's [7] main purpose is to determine a minimal set of cells and links, which is equivalent to the network generated by RCA. That is, all previously learned information should be retained in the pruned network. GAC attempts to determine a minimal set of connections, which may act as inhibitors of the information collector cells (ICC). Each information collector cell is introduced to the network as the result of an example in the training set which was distinct from all previously seen examples. GAC is completely described in [8].

2.1 Dynamic Plateau Modification of fuzzy membership functions

All fuzzy membership functions in SC-net are represented as trapezoidal fuzzy sets [7, 9]. They are represented in the network by a group of cells as shown in Figure 4 for the fuzzy variable teenager. Teenager takes membership values of 1 in [13..19], of course. In this implementation the membership goes linearly to 0 at the ages of 5 and 25. In the network ages are translated into [0,1] from the [0,100] year range. So the age of 22 is translated to 0.22. Figure 5 shows the actual graph of the membership function for the fuzzy teenager variable.

The dynamic plateau modification function (DPM) is designed to bring in the arms of the fuzzy membership function. In general, we allow the range of the membership function for unknown functions to initially be the range of the fuzzy variable. The range in which the function obtains a value of 1 is at least one point (all fuzzy sets in SC-net are normal in the sense that they contain at least one full member) and usually much smaller than the function range. Hence, for the teenage example with a 100 year range the right arm of the trapezoidal membership function would initially go to 0 at age 100, if we had no information on constructing the membership function other than where it is crisp (attains a membership value of 1). We always assume that the crisp (normal) portion of the membership function is known. The DPM function allows us to arbitrarily set the arms too wide and then adjust them during the learning process. Clearly, in our example it is impractical for someone 99 or 100 years old to have membership in the fuzzy set teenager.

A high-level description of the DPM method is as follows. When it is determined that the fuzzy membership value has caused an incorrect output, the maximal membership that will not cause an error is determined. This value for the set element given and the nearest element at which the membership function takes a value of 1 are used to specify the linear arm of the function. This provides a new upper or lower plateau value (point at which the function goes to 0) for the fuzzy membership function which is used to update the weights

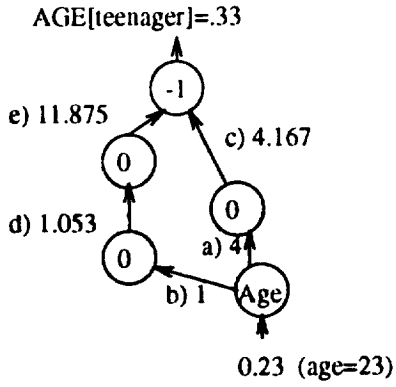


Figure 4: The fuzzy variable teenager.

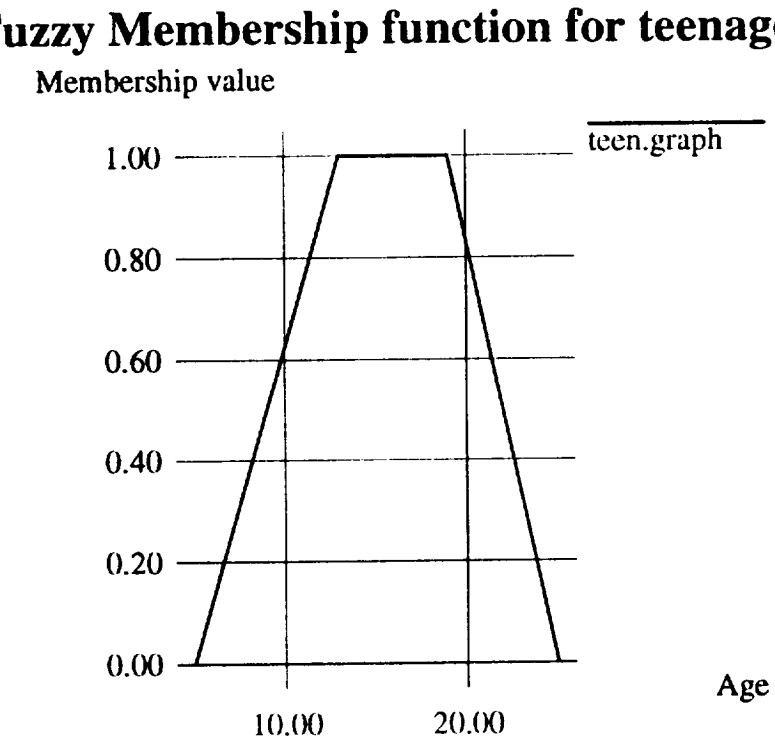


Figure 5: Graph of membership function for fuzzy variable teenager.

labeled a thru e in Figure 4 [9].

2.2 Automatic partition generator

In SC-net all real-valued inputs are modeled by a set of individual fuzzy sets which cover the range of the input. In the case that real-valued data is truly fuzzy, but domain experts do not exist to provide indications of how to model it by fuzzy sets, the choice of the fuzzy sets to cover the range is difficult. Since the data is fuzzy, it may not be possible to accurately identify distinct ranges of the real-valued output associated with specific output. However, this type of idea of associating (fuzzy) ranges with actual outputs can be used. The automatic partition generator (APG) is a method to develop a viable set of fuzzy sets for use in the learning process in domains which have real-valued input, but no expert identified ranges that may belong to specific fuzzy sets.

The APG algorithm works as follows. For each real-valued attribute or feature it makes a partition such that the boundary going from low value to higher value includes at least one element of a class. It will further contain as many elements of the same class as possible. Given the strategy to have all the partitions contain only one class, the maximum number of partitions for any given feature would be the number of classes and would indicate it is very difficult to partition the train set based on that feature or attribute alone. It is the case that a partition may be bounded on both sides by partitions that belong to the same class which is a different class than the examples in the bounded partition belong to.

3 The Nature of MRI Data

Magnetic Resonance Imaging (MRI) systems measure the spatial distribution of several soft tissue related parameters such as T1 relaxation (spin lattice), T2 relaxation (transverse) and proton density. By discrete variations of the radio frequency (RF) timing parameters, a set of images of varying soft tissue contrast can be obtained. The use of time varying magnetic field gradients provide spatial information based on the frequency or phase of the precessing

protons using both multi-slice (2DFT) or volume (3DFT) imaging methods [10, 11]. Hence, a multi-spectral image data set is produced.

In our work, male volunteers (25-45 years) and patient tumor studies were performed on a high field MRI system (1.5 tesla) using a resonator quaduature detector head RF coil. Transverse images of 5 mm thickness were obtained using a standard spin echo (SE) technique for T1 weighted images (pulse repetition time TR = 600 ms, echo time TE=20 ms) and proton density (ρ) and T2 weighted images (TR=3000 ms, TE=20 and 80 ms respectively), using the 2DFT multi-slice technique [12, 13, 2]. Volunteers were imaged for the same anatomical location.

Pixel intensity based classification methods were employed in this work as opposed to methods based on the calculation of magnetic resonance relaxation parameters. The latter methods require tailored RF pulse sequences [10, 11]. Image intensity based methods can be applied to any imaging protocol and are not restricted to the number of images acquired, i.e. it is possible to accommodate images with features other than MR relaxation parameters, such as perfusion and diffusion imaging, metabolic imaging and the addition of images from other diagnostic modalities [2]. The transverse images were acquired, centrally located in the resonator RF head coil, and hence did not require uniformity corrections for RF coil geometry or dielectric loading characteristics as developed at this institute [3]. Similarly, the subjects studied did not move significantly during the imaging procedure and hence, corrections were not required for related registration problems.

4 Segmenting magnetic resonance images

SC-net is a supervised instance-based learning system. Hence, in order to use it to segment an image a training set of labeled pixels must exist. Each pixel has 3 features associated with it a T1, T2 and proton density value. In this paper, we will focus on one normal slice and one abnormal slice. There are 271 pixels in the abnormal training set and 216 pixels in the normal training set. There are 5 classes in the normal train set; gray matter, white

matter, csf, fat and air. The abnormal train set also contains a class for tumor or pathology for a total of 6 classes. Each of the train sets was chosen by a radiological technician.

Each of the input features is real-valued taking values in [0,255] and hence will be represented as fuzzy sets within SC-net. However, it is unclear how these fuzzy sets should be constructed. Further, in [6] it is shown that the values associated with specific tissues vary from subject to subject with significant overlap. Therefore, the partitions of the input ranges for the initial fuzzy sets for each of the inputs were obtained by the use of the APG algorithm.

The inputs in each dimension are first translated from [0,max_value] $max\_value \leq 255$ to the [0,1] range. The APG algorithm is then run which, for example, in the normal (volunteer) training set produces 11 partitions in T1, 19 partitions in proton density (ρ) and 5 partitions in T2. It is interesting that T2 requires the least partitions as it has been the most used single parameter in the literature and few partitions will belong to features or attributes that are "good" data separators. The initial range of each constructed fuzzy set is [-0.2,1.2]. Allowing the range of the membership function to be larger than the range of the set it models is an implementation convention which allows membership values to be 1 at the edges of the actual range.

There are two possible ways to assign examples to classes. One is to use 5 outputs for the normal example and 6 outputs for the abnormal example. This is the most straightforward method. Another possibility exists, which is to use just 1 output. This output is then broken into 5 ranges for the normal example (i.e. [0,0.2], (0.2,0.4], (0.4, 0.6], (0.6, 0.8], and (0.8,1]) which respectively represent the 5 tissue types of interest. Similarly, the single output range can be broken up for 6 outputs. The use of one output provides a very compact network with just 3 inputs which fan out into 35 fuzzy sets in the normal example.

In all experiments, after training all of the remaining pixels are presented to the network for classification. The image is 256 by 256, which means that the training set is very small in relation to the total set of 65,536 pixels.

Table 1: Synthetic Colors for MR Tissue Classes.

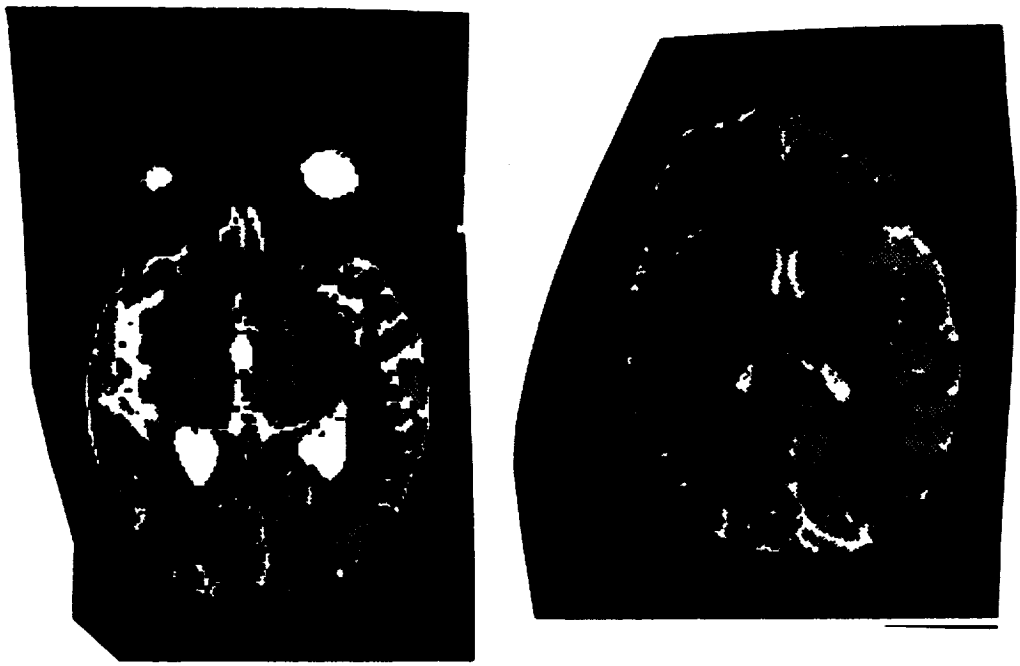
blue	air
yellow	cerebrospinal fluid (csf)
red	white matter
orange	gray matter
brown	fat
purple	pathology

4.1 Results

In Figure 6, we show the segmentation results for a patient with pathology (6a) using 6 outputs and a normal volunteer with 5 outputs (6b). In both cases the fuzzy outputs have been made into one crisp color. The chosen color is the one associated with the output which has the highest membership value. A color table for the figures is listed in Table 1. The patient with pathology has received chemo and radiation therapy which has eliminated obvious tumors, but left some pathology.

The segmentations in Figure 6 are comparable to segmentations pronounced as good by a team of radiologists [5]. The only real difference is that some fat (brown) shows up within the brain. However, this is a minor inconsistency. The case with pathology is segmented as well as any of the other fuzzy unsupervised and non-fuzzy supervised techniques used in [5]. In the lower left-hand part of the image the pathology is clearly defined and it can be seen that there is also pathology in the top of the image and the lower right-hand part of the image.

In Figure 7, we show the results using only 1 output for the abnormal case (7a) and normal case (7b). It can be seen that the segmentations are much the same as before. The fat in 7b is only weakly misclassified in this instance and barely shows up in the segmented image. These displays are fuzzy, which means that a pixel that strongly belongs to a class gets a bright color value, while a pixel that weakly belongs to a class is a darker shade of the same color. This generally shows the uncertainty in the segmentation better and tends



a)

b)

Figure 6: An abnormal and normal segmentation by SC-net with multiple outputs.

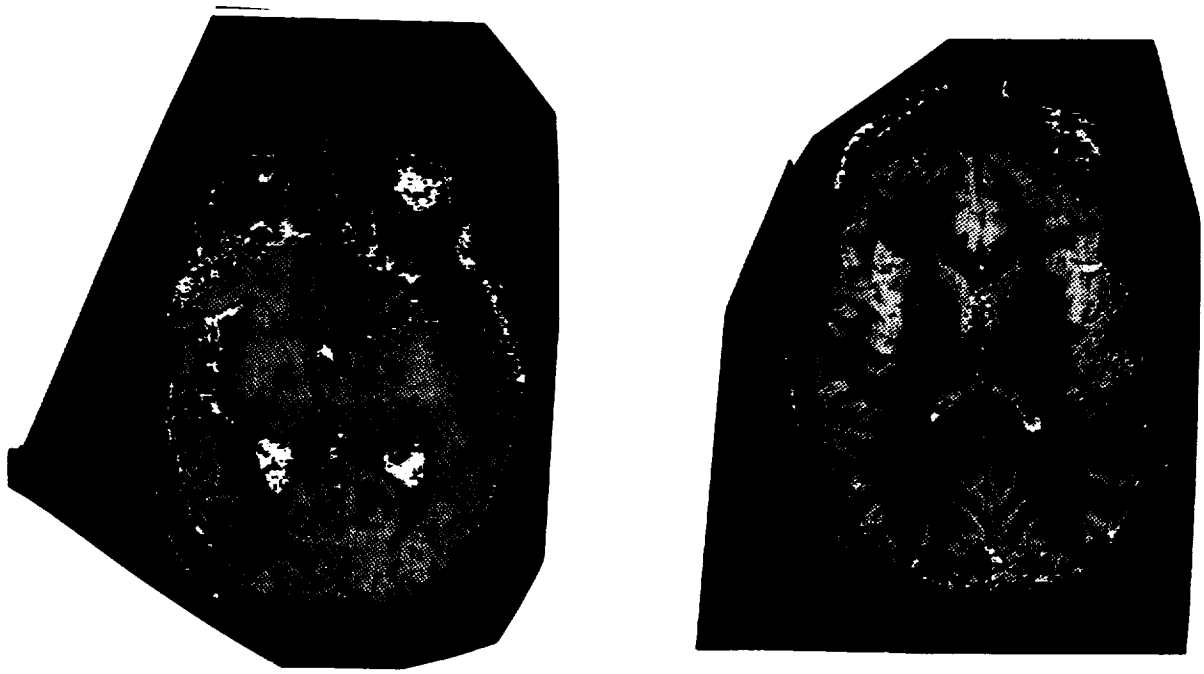
to highlight borders [5].

5 Summary

SC-net is able to provide good segmentations of MR images of the brain. This is a domain in which there is significant tissue overlap and the boundaries are fuzzy. With the use of the APG function the real-valued inputs are automatically partitioned into fuzzy sets. These fuzzy sets are further refined after the RCA learning algorithm has been applied by the use of DPM.

The results of the segmentation are comparable to those obtained by K nearest neighbor (K-nn) (K=7) and Cascade Correlation [5] in another study of supervised learning techniques. In the normal volunteer image the SC-net segmentation is a little clearer than the k-nn segmentation with the one exception of misclassified fat. The fuzzy connectionist representation of SC-net is very effective and fast in learning and classifying the MR images. The rules that are generated after the use of GAC for the normal case numbered 9 and 13 for the abnormal case. They can be used to provide a sense of what portions of which features are important in the recognition process. In Figure 8, the 9 rules for a normal case are shown. It can be seen that for output 5, fat, the 16^{th} partition of the T2 parameter is crucial. For output 2, csf, around the 2^{nd} proton density partition is the an important indicator. Output 1, which is air, is very easy to distinguish by one rule. This is a known fact since it essentially has a 0 return. The number of rules required to distinguish a class can also be an indication of how difficult it is to recognize. Hence, the rules can have semantic meaning and may be useful in tuning the system which is an advantage of a hybrid representation.

Acknowledgements: Thanks to Robert Velthuizen for helping us with the image display and providing an expert interpretation of the images. This research was partially supported by a grant from the Whitaker Foundation.



a)

b)

Figure 7: An abnormal and normal segmentation by SC-net with 1 output.

```
Rule 1:  if and( fuzzy(I3[p16]) = 1.000, fuzzy(I2[p12]) = 1.000,
fuzzy(I1[p5]) = 1.000 ) then Out5 ( 1.000 ).

Rule 2:  if and( fuzzy(I3[p16]) = 1.000, fuzzy(I2[p17]) = 1.000,
fuzzy(I1[p7]) = 1.000 ) then Out5 ( 1.000 ).

Rule 3:  if and( fuzzy(I3[p16]) = 1.000, fuzzy(I2[p19]) = 1.000,
fuzzy(I1[p16]) =  1.000 ) then Out5 ( 1.000 ).

Rule 4:  if and( fuzzy(I3[p2]) = 1.000, fuzzy(I2[p2]) = 1.000,
fuzzy(I1[p17]) = 1.000 ) then Out4 ( 1.000 ).

Rule 5:  if and( fuzzy(I3[p15]) = 1.000, fuzzy(I2[p3]) = 1.000,
fuzzy(I1[p17]) = 1.000 ) then Out3 ( 1.000 ).

Rule 6:  if and( fuzzy(I3[p22]) = 1.000, fuzzy(I2[p3]) = 1.000,
fuzzy(I1[p5]) = 1.000 ) then Out2 ( 1.000 ).

Rule 7:  if and( fuzzy(I3[p17]) = 1.000, fuzzy(I2[p2]) = 1.000,
fuzzy(I1[p5]) = 1.000 ) then Out2 ( 1.000 ).

Rule 8:  if and( fuzzy(I3[p19]) = 1.000, fuzzy(I2[p2]) = 1.000,
fuzzy(I1[p6]) = 1.000 ) then Out2 ( 1.000 ).

Rule 9:  if and( fuzzy(I3[p1]) = 1.000, fuzzy(I2[p1]) = 1.000,
fuzzy(I1[p1]) = 1.000 ) then Out1 ( 1.000 ).
```

Figure 8: Rules for normal volunteer

References

[1] Bensaid, A. M., Hall, L. O., Clarke, L. P., and Velthuizen, R. P. MRI Segmentation Using Supervised and Unsupervised Methods. IEEE Eng. Med. & Biol., Proceedings of 13th Annual Meeting, Oct. 31st-Nov. 3rd, Orlando, Florida, 1991.

[2] Clarke, L. P. Comparison of Bayesian Maximum Likelihood (MLM) and Artificial Neural Network for Supervised Tissue Classification and 3D Segmentation by MRI. Scientific Symposium, American Association of Physicists in Medicine (AAPM), 33rd Annual Meeting, San Francisco, California, July 1991; Med. Physics, 18(3), 673, 1991.

[3] Glennon, D. T., Clarke, L. P., Velthuizen, R. P. and Silbiger, M. L. MR Image Non-Uniformity Correction Techniques. IEEE Eng. Med. Biol., Proceedings of 13th Annual Meeting, Oct. 31st-Nov. 3rd, Orlando, Florida, 1991.

[4] Hall, L.O. and Romaniuk, S.G., " A hybrid connectionist, symbolic learning system", *AAAI-90*, Boston, Ma. 1990.

[5] Hall, L.O., Bensaid, A., Bezdek, J.C., Clarke, L., Silbiger, M. and Velthuizen, R. *A Comparison of Neural Network and Fuzzy Clustering Techniques in Segmenting Magnetic Resonance Images of the Brain*, IEEE Transactions on Neural Networks, V. 3, No. 5, 1992.

[6] Li, C., Goldgof, D. and Hall, L.O., *Towards Automatic Classification and Tissue Labeling of MR Brain Images*, International Association for Pattern Recognition Workshop on Structural and Syntactic Pattern Recognition, Bern, Switzerland.

[7] Romaniuk, S.G. " Extracting Knowledge from a Hybrid Symbolic, Connectionist Network", Ph.D. Dissertation, Department of Computer Science and Engineering, University of South Florida, 1991.

[8] Romaniuk, S.G. and Hall, L.O. "SC-net: A Hybrid Connectionist, Symbolic Network", *Information Sciences*, To Appear, 1993.

[9] Romaniuk, S.G. and Hall, L.O. "Learning Fuzzy Information in a Hybrid Connectionist, Symbolic Model", *IEEE International Conference on Fuzzy Systems 1992*, pp. 309-312., San Diego, Ca. 1992.

[10] Bottomley, P. A., Foster, T. H., Argersinger, R. E. and Pfeifer, L. M. A. Review of Normal Tissue Hydrogen NMR Relaxation Times and Relaxation Mechanisms from 1-100 MHz: Dependence on Tissue Type, NMR Frequency, Temperature, Species, excision and Age. Medical Physics, 11(4), 425-448, 1984.

[11] Hyman, T. J., Kurland, R. J., Leroy, G. C. and Shoop, J. Characterization of Normal Brain Tissue Using Seven Calculated MRI Parameters and a Statistical Analysis System. Mag. Res. Med. 11, 22-34 1989.

[12] Bensaid, A. M., Hall, L. O., Clarke, L. P., and Velthuizen, R. P. MRI Segmentation Using Supervised and Unsupervised Methods. IEEE Eng. Med. & Biol., Proceedings of 13th Annual Meeting, Oct. 31st-Nov. 3rd, Orlando, Florida, 1991.

[13] Schellenberg, J. D., Naylor, W. C. and Clarke, L. P. Application of Artificial Neural Networks for Tissue Classification from Multispectral Magnetic Resonance Images of the Head. IEEE Symposium, Computer-Based Medical Systems, Third Annual Conference Proceedings, Chapel Hill, North Carolina, June 3-6, 1990.

On the Neural Substrates Leading to the Emergence of Mental Operational Structures

H. Öğmen

Department of Electrical Engineering
University of Houston
Houston, Texas, 77204-4793

ABSTRACT

A developmental approach to the study of the emergence of mental operational structures in neural networks is presented. Neural architectures proposed to underlie the six stages of the sensory-motor period are discussed.

1 Introduction

Historically, the study of intelligence has been polarized into explanations based on neurophysiology and those based on logic. The same dichotomy manifests itself in current approaches, the former corresponding to the neural network theory and the latter to the symbolic reasoning schemes used in artificial intelligence (logic, fuzzy logic, etc.). At a first glance these two explanations fail to extend to each other, for logic does not tell us anything about neurophysiology and it seems difficult to explain the rules of logic from the connectivity and firing patterns of neurons. However, logic is housed in neurophysiological substrates and there should be a reconciliation between these two explanations (if we reject dualism). Since logical reasoning emerges as a result of an extensive development and since the early phases of development consist of simpler behaviors, the link between adult intelligence and the underlying neural correlates can be established by relating the early developmental stages to neurophysiological substrates and by studying the adaptive dynamics of the system that leads to the emergence of higher mental operations. The theory of genetic psychology (Piaget, 1967) provides us with a very detailed study of various phases starting from birth to the adulthood. This paper extends various psychological concepts of the theory of genetic psychology to the neurophysiological domain. In particular, it outlines neural networks proposed to give rise to the various stages of the sensory-motor period.

2 A neural theory of development

Our study of development is closely linked to the theory of genetic psychology (Piaget, 1967). Piaget (1963) named the post-natal developmental period where language is absent the *sensory-motor period* and suggested the existence of six consecutive stages that govern its dynamics. These stages start from reflexes and end with mental manipulations of sensory-motor schemes to invent new intelligent structures. Mental internalization of early sensory-motor schemes lead to the capability of applying them to formal reasoning in the adult life.

2.1 The six stages of the sensory-motor level according to the theory of genetic psychology

The first stage of the sensory-motor period consists of simple reflexive sensory-motor behavior. In the second stage, the repetitive use of these reflexes, called "primary circular reactions", leads to the formation of habits. The primary circular reactions refine genetically encoded reflexes and enable the emergence of multi-modal coordination. At the third stage the infant starts to draw a distinction between the "means" and "ends" and uses "secondary circular reactions". During the fourth stage, existing sensory-motor schemes are coordinated and extended to new situations. In stage 5, new sensory-motor schemes are acquired through physical groping. Finally during the sixth stage, which marks the end of the sensory-motor level, new sensory-motor schemes are acquired by mental groping.

2.2 A model for primary sensory-motor schemes

We first start by describing the seed sensory-motor circuit with nonassociative learning properties (Öğmen 1991, Öğmen & Moussa, in press). The architecture, which is illustrated in Figure 1, has three main parts: the sensory circuit, the sensory-motor gate circuit, and the motor circuit. Since the model was originally formulated explicitly for the prototypical landing behavior of the fly, the sensory and motor parts are specialized for this animal. The sensory part consists of visual signals conveyed by the compound eyes and of tactile pathways. In the fly, signals from the compound eyes are processed by three optic ganglia: lamina, medulla, and lobula complex, denoted by La, Me, Lo respectively in Figure 1. The output of the visual processing stage results from a behavior sensitive pooling of motion detector neuron activities. In the case of landing, stimuli indicating the approach of a landing site, such as expanding patterns, are detected by an appropriate pooling of directionally selective large field motion sensitive neurons and constitute the "agonist" input to the sensory-motor gate circuit. This input is denoted by V_{ag} in Figure 1. Stimuli of opposite character (such as contracting stripes) constitute the "antagonist" input denoted by V_{an}. Agonist and antagonist tactile inputs are added to these visual signals. One such input is marked by I_{ta} in Figure 1. The following stage, which is a gated dipole anatomy (Grossberg, 1972), constitutes the *sensory-motor gate* network because this is the stage where various sensory signals are pooled to determine whether a motor command signal will be issued. The sensory-motor gate network has agonist and antagonist outputs denoted respectively by x_{ag} and x_{an}. These signals project to motor circuits (not shown) to control agonist-antagonist muscle pairs (indicated by Ag and An). This sensory-motor model exhibits nonassociative learning as observed in the landing reaction of the fly as well as in human infants (Lipsitt 1990).

Figure 2 illustrates this architecture augmented with adaptive capabilities. The adaptive version of this sensory-motor model has also three major parts: sensory, sensory-motor gate, and motor. These parts are augmented by inclusion of adaptive mechanisms similar to those proposed in the INFANT (Kuperstein & Rubinstein 1989) and AVITE (Gaudiano & Grossberg 1991) models. x_e represents an environmental variable. These environmental variables are converted into neural activities by the sensory loci. The activities of the sensory loci are denoted by x_s. The sensory loci project to the sensory-motor gate networks. The first layer of the sensory motor gate network consists of nodes interconnected by recurrent on-center off-surround connections. Each node corresponds to the agonist-antagonist inputs of a given sensory-motor scheme. The gated dipole of Figure 1 is represented in a condensed way by a single node in this layer. An arbitrary number of sensory-motor schemes, similar to the one shown in Figure 1, exists. The competition between these nodes selects and triggers one of the sensory-motor schemes. The second layer of

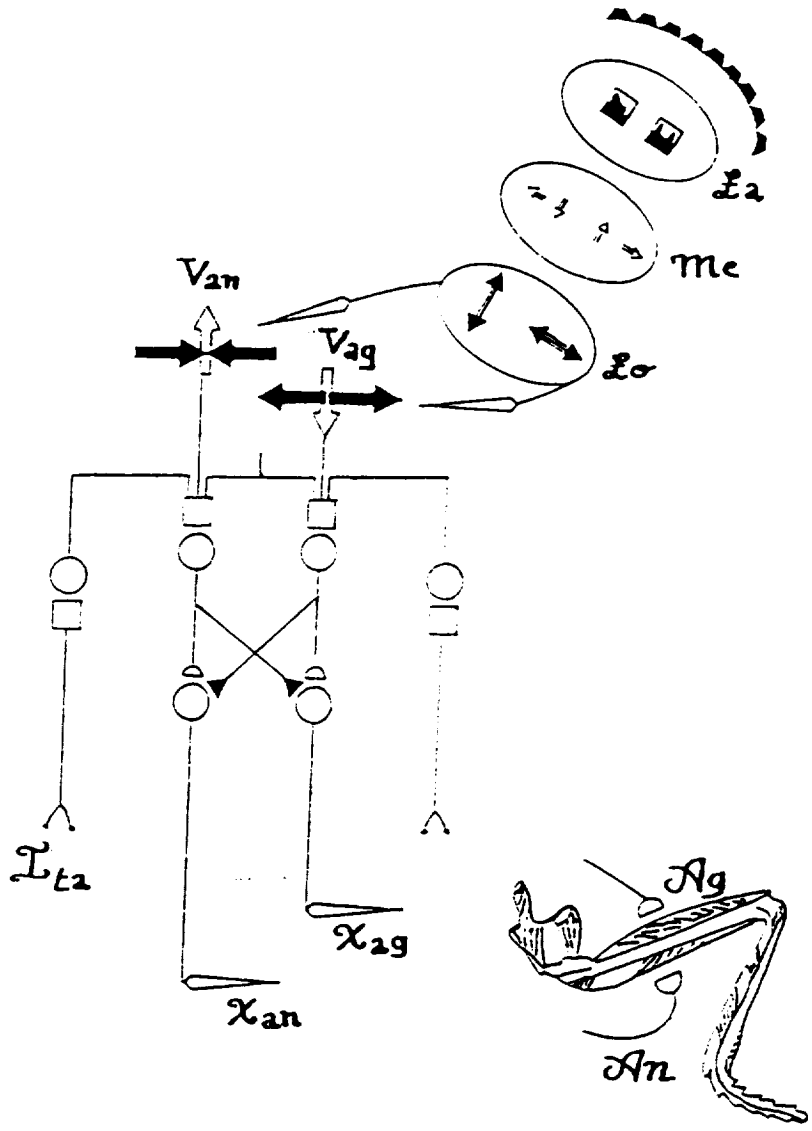


Figure 1: Model for primary sensory-motor schemes (Ögmen & Moussa in press).

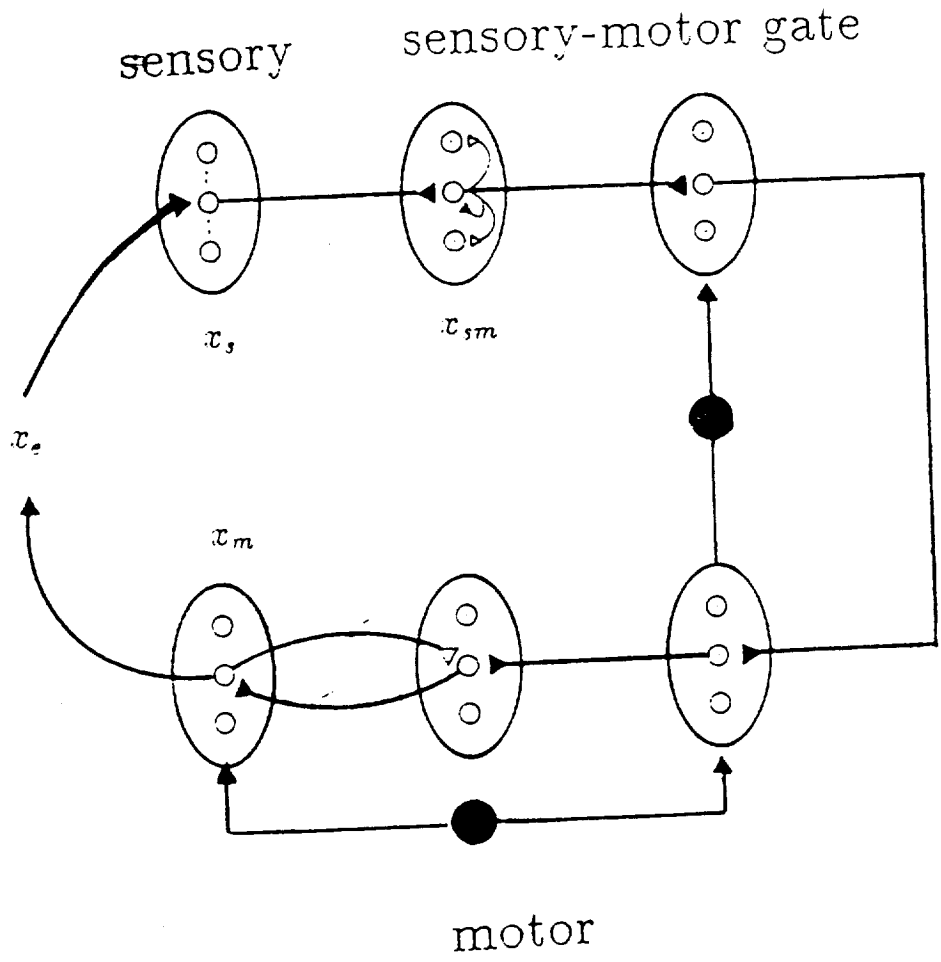


Figure 2: Adaptive version of the sensory-motor model shown in Figure 1.

the sensory-motor gate part is is used convert the representation of the sensory space into the representations of the motor space. The connections converging to this layer are adaptive. The motor part consists of three layers and has AVITE model's anatomy (Gaudiano & Grossberg 1991). The output layer produces motor command signals while the input layer receives the desired motor command signals. The layer between these "target" and "present" motor signals computes the error. This organization constitutes a basic feedback control system where the output is driven by an error signal during real-time operation. Moreover, the same "motor error" layer is used to adapt to changes in the plant, making the system an adaptive feedback control system. What is notable in this anatomy is that the same error signals are used both for performance and learning. Another important feature of the AVITE circuit is the "Endogenous Random Generator" (ERG) which generates random postures and enables the spontaneous learning of sensory-motor coordinate transforms. The filled circles depict in a condensed way the function of the ERG. During the active phase of the ERG a random motor signal is dictated to the output of the motor circuit. During the passive phase of the ERG, activity of the motor layer is transferred to the first layer of the motor circuit which is transferred to the buffer layer of the sensory-motor gate network. The arrow from the bottom filled circle to the motor output layer depicts the generation of random motor signals. The pathway from the motor signal level to the motor target level depicts the transfer of activity between these layers during the learning phase. A similar transfer occurs between the motor target layer and the sensory-motor gate buffer layer. Note that, in addition to internal feedback, the sensory-motor circuit constitutes a closed loop through the environmental variable denoted by x_e. The interaction between the environmental variables and neural variables is essential and constitutes an overall organization by the relationship of *assimilation* that unites them (Piaget, 1963).

The circuit described above constitutes the proposed basic neural correlate for the reflexive behavior of Stage 1. The adaptive nature of the circuit requires the use of these reflexes for consolidation and fine tuning. The repetitive use of reflexes leads to the second stage, the stage of habits. Figure 3 shows how the beginnings of "cortexification" occurs at this stage. In Figure 3, the architecture of Figure 2 is depicted in a simplified form as a closed loop of environmental, sensory, sensory-motor gate, and motor variables denoted by x_e, x_s, and x_m respectively. To this basic loop, additional circuits, proposed to be of cortical origin, are added. These cortical networks receive sensorial inputs and have a feedback structure. They are proposed to be adaptive resonance theory (ART) architectures (Carpenter & Grossberg 1988, Grossberg 1976) and each node represents a layer of ART. At this stage, sensorial stimuli start to be recorded and generalized cortically. The outputs of these ART circuits also make connections with the sensory-motor circuits. The connections from ART circuits to sensory-motor loops are proposed to be Hebbian synapses so that an association between the cortical representation of sensory stimuli and the active sensory-motor circuits occurs through the reinforcement of the synaptic weight between the ART circuits and the sensory-motor gate nodes. A second feature of cortical development at this stage is the beginnings of cortico-cortico associations. This is shown by synaptic connections between various ART circuits. This phase of operation corresponds to Stage 2, in that the sensory-motor schemes are not decomposed and consequently there is no differentiation between the "means" and the "ends". The lack of decomposition comes from the property that one cannot, at this point, access these closed loop circuits from an arbitrary entry point and the only way to activate them is to provide the appropriate environmental signals.

During Stage 3, such a decomposition is introduced by the development of "secondary circular reactions". Once the sensory-motor repertoire becomes sufficiently rich and multi-modal coordination (in particular the coordination between vision and grasping) reaches a satisfactory level,

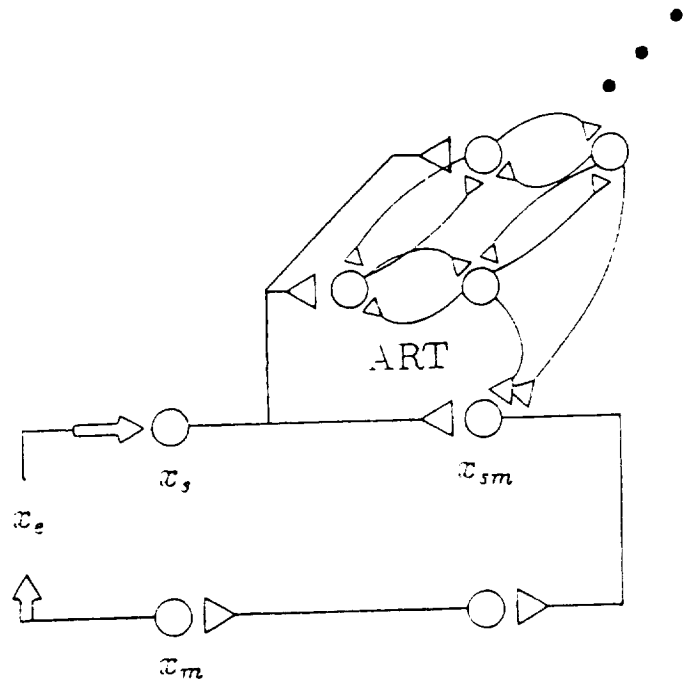


Figure 3: The architecture of Figure 2 is represented in a simplified form as a closed loop of environmental variables x_e, sensory variables x_s, sensory-motor gate variables x_{sm}, and motor variables x_m. To these basic loops additional circuits are added to explain the beginnings of "cortexification".

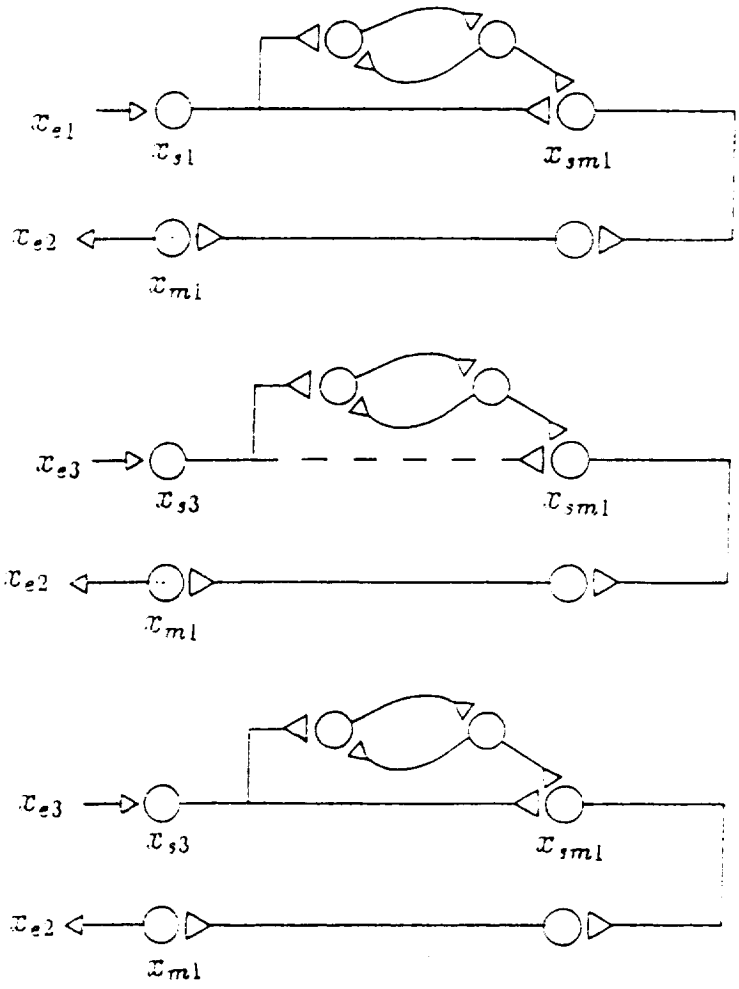


Figure 4: Secondary circular reactions

the infant starts grasping objects placed in her vicinity. Assume that the infant is presented with a toy, grasps the toy, shakes the toy, and that the toy produces a characteristic sound new to the infant. This sound will retrigger the sensory-motor scheme, i.e. will generate a circular reaction. Later, when the infant is presented with a similar toy, she will try the same sensory-motor scheme. Similarly, when the sound of the toy is reproduced by the experimenter, she will look for the toy (Piaget 1963).

Figure 4 illustrates how this occurs in the proposed neural circuits. Initially, a toy is presented (x_{e1}) and it triggers a sensory-motor scheme (grasping and shaking) whose output is a surprising sound (x_{e2}). The sensory-motor scheme is repeated many times until the novelty of stimuli vanishes. [1] During this circular reaction, a connection from the cortical circuits to the sub-cortical circuits is also reinforced. If there are consistent stimuli pairings, association between these stimuli also occurs. In the example of toy shaking, at least an association between the object (x_{e1}) and the sound (x_{e2}) occurs. Later, assume that the infant is presented with a different object x_{e3} and that this object does not trigger directly the grasping scheme (the dotted connection in Figure 4). Now, if this object has an equivalent cortical generalization, it will send a signal to this sensory-

[1] In the basic architecture of Figure 2, habituating sensory and sensory-motor gate signals underlie the habituation properties of the circuit.

motor scheme via the cortical connection which was reinforced during the circular reactions. If this signal is strong enough to trigger the sensory-motor scheme without the sensory signal, the infant will shake this new object. If this object also makes a similar sound ($x_{\ell 2}$), it will re-activate this sensory-motor scheme via a circular reaction. The repetition of the sensory-motor scheme will reinforce the direct sub-cortical connection as shown at the bottom of Figure 4.

While the previous example considered a single sensory-motor scheme for simplicity, in practice a larger number of environmental variables (the first being the object, the last being the sound) are involved.

Note that one can see the important distinction that Piaget drew between his notion of adaptation through an equilibrium between assimilation and accommodation and the associationistic theory of intelligence. While, the behavioral analysis of the network described above may look like an associationistic paradigm, it is important to emphasize that the associations are not passive but involve existing sensory-motor schemes. Initially, $x_{\ell 1}$ and the intermediate environmental variables are assimilated to generate this particular sensory-motor scheme. During this assimilation process, other environmental variables are also registered. This registration is primed by the neural activities occurring in the sensory-motor scheme that is active. The time scales of association (i.e. the inter-stimulus intervals) are not arbitrary but are determined by the temporal characteristics of the active sensory-motor scheme. Later, when $x_{\ell 3}$ is represented to the system, it can activate the same sensory-motor scheme either because it is an equivalent stimulus or because it was associated with $x_{\ell 1}$ during the previous repetitions. Consider the classical example of food bell pairings. The presentation of an object that resembles food will trigger a complex sensory-motor scheme in a dog. The same sensory-motor scheme can also be generated by a different food. Another possibility is that a conditioned stimulus, such as the sound of a bell, is delivered. In a future trial, if this bell is delivered alone it will trigger the sensory-motor scheme. However, following the first phase (salivation) if the appropriate environmental variables cannot be assimilated, the chain of actions will be broken and the global sensory-motor scheme will be aborted (the dogs do not continue to chew etc..). If such a pairing ceases to occur, it will be extinguished. However, assume now that the conditioned stimulus not only generates the initial stages of the sensory-motor scheme but also is accompanied by the appropriate intermediary environmental variables so that a complete assimilation cycle can occur. The successful completion of the assimilation will reinforce the direct path so that this stimulus will be able to generate the whole cycle directly. This change, which consists of an adaptation to the environment, is called *accommodation* (Piaget, 1963).

Once the sensory-motor schemes become accessible via cortical circuits as outlined above, they become decomposed and the essential property of Stage 3 emerges: the distinction between means and ends. A sensory-motor scheme is the means to achieve a goal. The goal corresponds to the final environmental variable or internal variables associated with this external variable. Assume that the infant acquired a sensory-motor scheme consisting of opening a drawer and receiving a new toy. Initially, this sensory-motor scheme can be generated only by the sight of the drawer. Once it becomes decomposed, the infant will look for a drawer with the goal of receiving a new toy.

In stage 4, the decomposed sensory-motor schemes are coordinated into new wholes to achieve new goals. The activation of a new goal can generate multiple or a chain of existing sensory-motor schemes. In the case of multiple activation, the competition at the sensory-motor gate level selects the best alternative. In the case of a chain, the chain gets activated as a whole as long as the appropriate environmental variables can be assimilated. This new grouping becomes reinforced if it leads to the achievement of the goal and can be accessed more readily in the future.

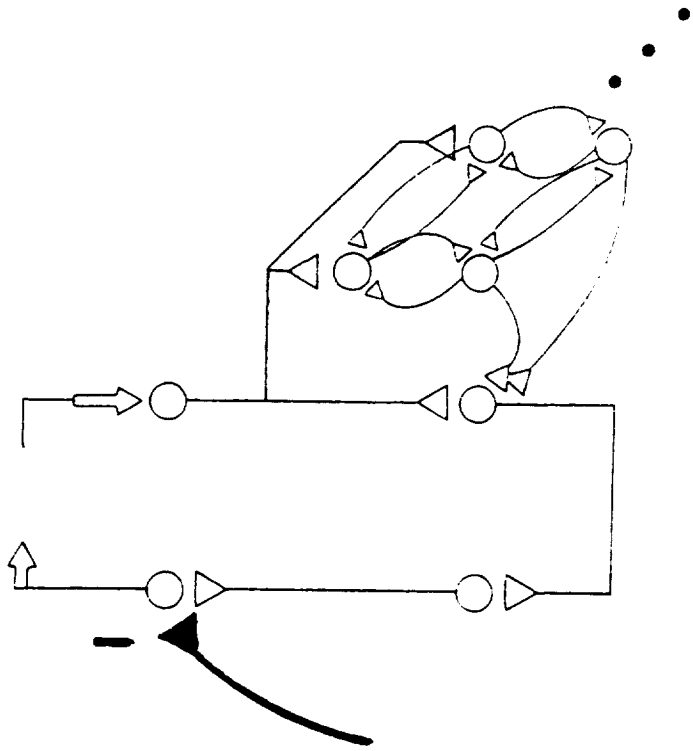


Figure 5: Mental groping: Motor layers are actively inhibited and the infant intentionally activates sensory-motor schemes without motor reactions.

If the existing sensory-motor schemes fail to combine with the environmental variables in order to complete the cycle leading to the achievement of the goal, new means have to be invented. This occurs in Stage 5 by use of physical groping. Assume that the goal activates some existing sensory-motor schemes through the competition at the sensory-motor gate layer. This competition will start with the best possibility and the system will try different schemes. If all fails, the ERG will generate random motor behavior or will bias existing sensory-motor schemes. This constitutes the basis of physical groping. If by chance the resulting behavior reaches the goal, it will be reinforced through circular reactions and will be integrated to the sensory-motor repertoire as a newly discovered means.

The functional operation of Stage 6 is illustrated in Figure 6. At this stage, the motor layers are actively inhibited. This way, the discovery of new means can be disconnected from actual physical action. As a result, the subject can carry out "mental groping" and discover new means without physical contact.

3 Concluding remarks

The early developmental period outlined above indicate that passive perception is inadequate for explaining intelligence and the system should actively explore the environment to generate a rich repertoire of sensory-motor schemes whose abstractions lead to the formal reasoning structures in adult life.

Acknowledgment: This work was supported in part by a grant from NASA-JSC.

REFERENCES

Carpenter GA, Grossberg S (1988) The art of adaptive pattern recognition by a self-organizing neural network. IEEE Computer, pp. 77-88.

Gaudiano P, Grossberg S (1991) Vector associative maps: Unsupervised real-time error-based learning and control of movement trajectories. Neural Networks. 4:147-183.

Grossberg S (1972) A neural theory of punishment and avoidance. II. Quantitative theory. Mathematical Biosciences 15:253-285

Grossberg S (1976) Adaptive pattern classification and universal recording II: Feedback, expectation, olfaction, and illusions. Biol Cybern. 23:187-202.

Kuperstein M, Rubinstein J (1989) Implementation of an adaptive neural controller for sensory-motor coordination. IEEE Control Systems. 9:25-30.

Lipsitt LP (1990) Learning processes in the human newborn: Sensitization, habituation, and classical conditioning. In: Diamond A (ed.) The development and neural bases of higher cognitive functions. Annals of the New York Academy of Sciences. (vol 608)

Öğmen H, Moussa M (in press) A neural model for nonassociative learning in a prototypical sensory-motor scheme: The landing reaction in flies. Biological Cybernetics.

Piaget J (1963) *The origins of intelligence in children.* Norton: New York.

Piaget J (1967) *La psychologie de l'intelligence.* Armand Colin: Paris.

A Fuzzy Controller with Nonlinear Control Rules Is the Sum of a Global Nonlinear Controller and a Local Nonlinear PI-like Controller

Hao Ying

1) Department of Physiology and Biophysics
2) Biomedical Engineering Center
3) Office of Academic Computing
University of Texas Medical Branch
Galveston, TX 77555, USA.

Abstract

The fuzzy controllers studied in this paper are the ones that employ N trapezoidal-shaped members for input fuzzy sets, Zadeh fuzzy logic and a centroid defuzzification algorithm for output fuzzy set. The author analytically proves that the structure of the fuzzy controllers is the sum of a global nonlinear controller and a local nonlinear proportional-integral-like controller. If N approaches ∞, the global controller becomes a nonlinear controller while the local controller disappears. If linear control rules are used, the global controller becomes a global two-dimensional multilevel relay which approaches a global linear proportional-integral (PI) controller as N approaches ∞.

1. Introduction

Efforts have been made to clarify the fuzzy controller structures. The structure of a nonlinear fuzzy controller was revealed using a novel method (Ying, 1987; Ying et al., 1990). The work showed that a simplest possible nonlinear fuzzy controller was equivalent to a nonlinear PI controller. In (Ying, 1991), the author analytically proved that the structure of a typical nonlinear fuzzy controllers with linear fuzzy control rules is the sum of a global two-dimensional multilevel relay and a local nonlinear PI controller. The author makes further efforts in this paper to investigate the structure of fuzzy controllers using any type of fuzzy control rules, covering a much broader range of fuzzy controllers.

2. Analytical Analysis of the Structure of the Fuzzy Controllers

2.1 Components of the Fuzzy Controllers

A fuzzy controller usually employs error and rate change of error (rate, for short) about a setpoint as its inputs. That is

$$e^* = GE \cdot e(nT) = GE[y(nT) - \text{setpoint}] \quad (2.1)$$

$$r^* = GR \cdot r(nT) = GR[e(nT) - e(nT-T)] \quad (2.2)$$

where e(nT), r(nT) and y(nT) designate error, rate, and process output at sampling time nT (T is sampling period), respectively. Error at sampling time (n-1)T is specified as e(nT-T). The setpoint is the desired target of the process output and GE and GR are the scalars for the error and rate.

Input fuzzy sets, "error" and "rate," are obtained by fuzzifying e^* and r^*. Assume there are J ($J \geq 1$) members for positive "error" ("rate"), J members for negative "error" ("rate") and one member for zero "error" ("rate"). Therefore, there are total

$$N = 2J+1 \tag{2.3}$$

members for the fuzzy set "error" ("rate"). Members of "error" ("rate") are represented as E_i (R_i) where $-J \leq i \leq J$. The membership functions corresponding to these members are denoted as $\mu_i(x)$ which has a central value λ_i. Define $\lambda_{-J}=-L$, $\lambda_0=0$, and $\lambda_J=L$. Let the space between the central values of two adjacent members be equal. Then the space, denoted as S, is:

$$S = \frac{L}{J} \tag{2.4}$$

and consequently the central value of $\mu_i(x)$ is $\lambda_i=i \cdot S$.

The $\mu_i(x)$ in this study is the commonly-used trapezoidal-shaped membership function. Assume the membership functions for "error" and "rate" are identical, and specifically denote $\mu_i(e^*)$ as the membership function for E_i and $\mu_i(r^*)$ as the membership function for R_i. The trapezoidal-shaped membership function $\mu_i(x)$ satisfies the following two conditions:

(1) For $-J+1 \leq i \leq J-1$, (2.5)

$$\mu_i(x) = \begin{cases} 0, & x < (i-1)S \\ \frac{1}{S-A}[x-(i-1)S], & (i-1)S \leq x \leq i \cdot S - A \\ 1, & i \cdot S - A \leq x \leq i \cdot S + A \\ -\frac{1}{S-A}[x-(i+1)S], & i \cdot S + A \leq x \leq (i+1)S \\ 0, & x > (i+1)S \end{cases}$$

(2) For $i = J$ or $i=-J$,

$$\mu_J(x) = \begin{cases} 0, & x < (J-1)S \\ \frac{1}{S-A}[x-(J-1)S], & (J-1)S \leq x \leq J \cdot S - A \\ 1, & J \cdot S - A \leq x < +\infty \end{cases}$$

$$\mu_{-J}(x)=\begin{cases}1, & -\infty < x \le -J\cdot S + A \\ -\dfrac{1}{S-A}[x-(-J+1)S], & -J\cdot S + A \le x \le (-J+1)S \\ 0, & x > (-J+1)S.\end{cases}$$

An illustration of the definition is given in Fig. 1.

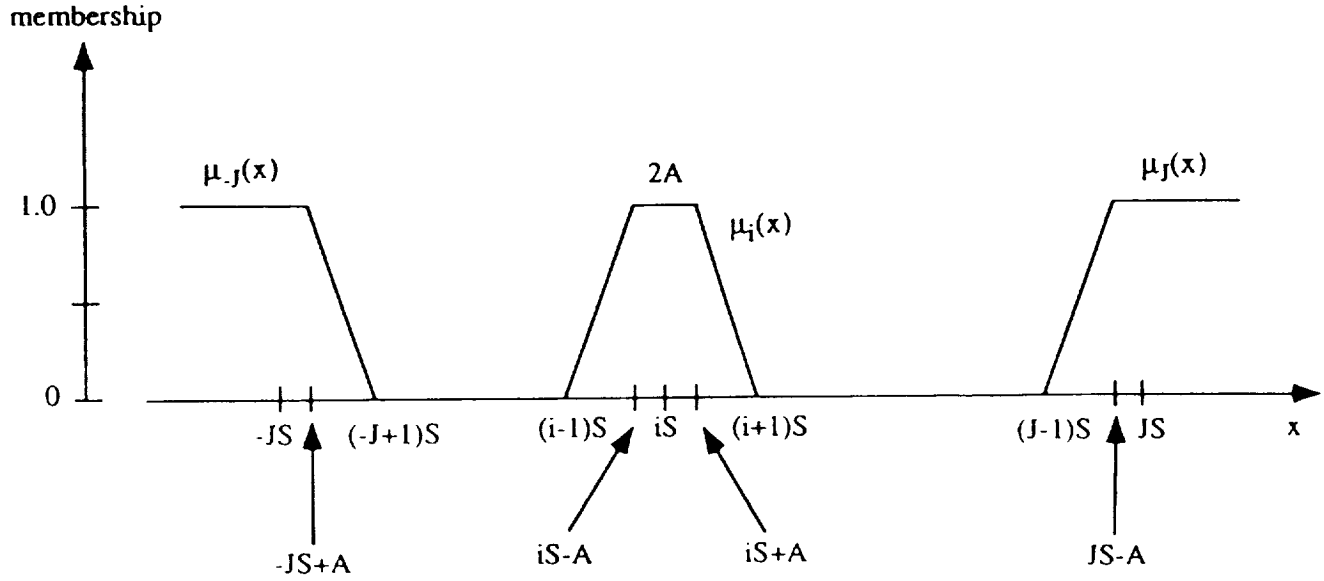


Figure 1. Illustration of the definition of the trapezoidal-shaped membership function.

Denote U_k as a member of the output fuzzy set "incremental output" ("output," for short) and assume there are

$$M = 2K + 1 \tag{2.6}$$

such members where

$$K = \text{Max}\{|f(i, j)|\}. \tag{2.7}$$

f will be described below. The central values of the members of the fuzzy set "output" are designated as γ_k ($-K \le k \le K$) and let γ_{-K}=-H, γ_0=0 and γ_K=H. Further, let the space between the central values of two adjacent members be equal. Consequently, the space, denoted as V, is

$$V = \frac{H}{K} \tag{2.8}$$

and the central value of a member of "output," U_k, can be written as $\gamma_k = k\cdot V$. The membership functions of "output" are required to be regular, unimodal and symmetrical about its central value γ_k. The shape of the membership functions of all the members is identical.

N^2 fuzzy control rules are constructed according to the following rule:

IF "error" is E_i and "rate" is R_j THEN "output" is U_k (2.9)

where $k = f(i, j)$. f, determined by the constructors of the fuzzy controllers, may be any function as long as its value is always an integer with respect to the inputs, i and j, because the index k must be an integer.

Zadeh fuzzy logic AND is used to execute the IF side of the fuzzy control rule in (2.9). That is,

$$\mu(i, j) = \text{Min}(\mu_i(e^*), \mu_j(r^*)) \tag{2.10}$$

where $\mu(i, j)$ is the membership for the member U_k obtained when E_i and R_j are in the IF side. The center of gravity of defuzzification algorithm is used. The scaled crisp incremental output, denoted as $GU \cdot \Delta u(nT)$, is calculated as

$$GU \cdot \Delta u(nT) = GU \frac{\sum \mu(i, j)\gamma_k}{\sum \mu(i, j)} \tag{2.11}$$

where GU is the scalar for the incremental output.

2.2 Main Results

Theorem 1.

The structure of the fuzzy controllers, constructed by the components defined in the above section, is the sum of a global nonlinear controller (denoted as $\Delta u_G(i, j)$) and a local nonlinear PI-like controller (denoted as $\Delta u_L(i, j)$).

Proof.

Without losing generality, assume that,

$$\begin{aligned} &iS \le e^* \le (i+1)S \\ &jS \le r^* \le (j+1)S. \end{aligned} \tag{2.12}$$

$\mu_i(e^*)$, $\mu_{i+1}(e^*)$, $\mu_j(r^*)$ and $\mu_{j+1}(r^*)$, which are the respective memberships for the members E_i, E_{i+1}, R_j and R_{j+1}, are obtained by fuzzifying e^* and r^*. Membership for all other members of "error" and "rate" is zero. Therefore, only the following four fuzzy control rules are executed:

If "error" is E_{i+1} and "rate" is R_{j+1} then "output" is U_{k1} (r1)
If "error" is E_{i+1} and "rate" is R_j then "output" is U_{k2} (r2)
If "error" is E_i and "rate" is R_{j+1} then "output" is U_{k3} (r3)
If "error" is E_i and "rate" is R_j then "output" is U_{k4} (r4)

where

$k_1 = f(i+1, j+1)$, $k_2 = f(i+1, j)$, $k_3 = f(i, j+1)$ and $k_4 = f(i, j)$.

Applying the equation (2.10) to each of the fuzzy control rules, we get

$$\mu(i+1, j+1) = \text{Min}(\mu_{i+1}(e^*), \mu_{j+1}(r^*)) \quad (r1^*)$$
$$\mu(i+1, j) = \text{Min}(\mu_{i+1}(e^*), \mu_j(r^*)) \quad (r2^*)$$
$$\mu(i, j+1) = \text{Min}(\mu_i(e^*), \mu_{j+1}(r^*)) \quad (r3^*)$$
$$\mu(i, j) = \text{Min}(\mu_i(e^*), \mu_j(r^*)). \quad (r4^*)$$

In order to decide the outcomes of the Min operations in $(r1^*)$ to $(r4^*)$, the author configures a square, which has 16 regions in it as shown in Fig. 2. In different regions, $\mu_i(e^*)$, $\mu_{i+1}(e^*)$, $\mu_j(r^*)$ and $\mu_{j+1}(r^*)$ have different relationships in terms of their magnitudes and consequently the Min operations in $(r1^*)$ to $(r4^*)$ can be evaluated. For example, in region IC3, the following inequalities can be obtained: $\mu_i(e^*) \geq \mu_j(r^*)$, $\mu_{i+1}(e^*) \leq \mu_{j+1}(r^*)$, $\mu_i(e^*) \leq \mu_{j+1}(r^*)$ and $\mu_j(r^*) \leq \mu_{i+1}(e^*)$. As a result, $\mu(i+1, j+1) = \mu_{i+1}(e^*)$, $\mu(i+1, j) = \mu_j(r^*)$, $\mu(i, j+1) = \mu_i(e^*)$ and $\mu(i, j) = \mu_j(r^*)$, based on $(r1^*)$ to $(r4^*)$. Similarly, $(r1^*)$ to $(r4^*)$ for the rest of 15 regions can be evaluated.

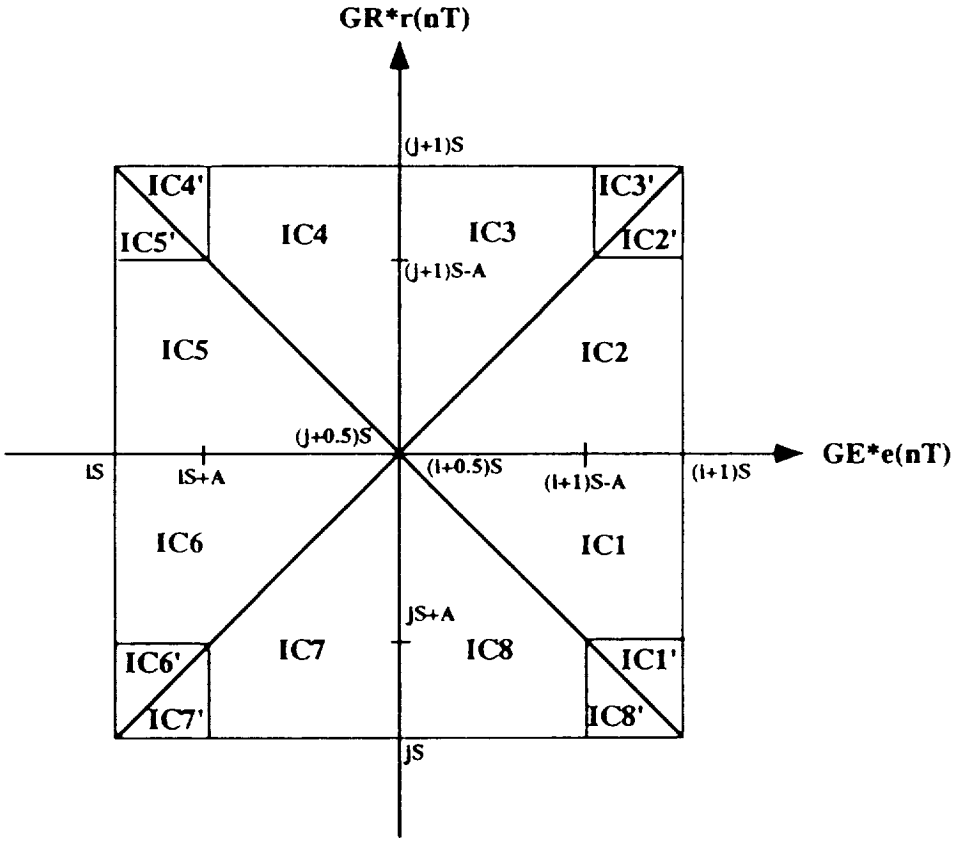


Figure 2. Possible input combinations (IC) of scaled error, e^* (GE·e(nT)), and scaled rate change of error, r^* (GR·r(nT)), of process output when both e^* and r^* are within the interval [-L, L].

Substituting these outcomes into the defuzzification algorithm (2.11) and simplifying the resulting expression, GU·Δu(nT) for the 16 regions can be found. To illustrate this procedure more clearly , let us take region IC3 again as an example.

Substituting μ(i+1, j+1), μ(i+1, j), μ(i, j+1) and μ(i, j) for the IC3 region into (2.11),

$$\begin{aligned} GU \cdot \Delta u(nT) &= \frac{k_1\mu_{i+1}(e^*)+k_2\mu_j(r^*)+k_3\mu_i(e^*)+k_4\mu_j(r^*)}{\mu_{i+1}(e^*)+\mu_j(r^*)+\mu_i(e^*)+\mu_j(r^*)} V \cdot GU \\ &= k_3 \cdot V \cdot GU + \frac{(k_1-k_3)\mu_{i+1}(e^*)+(k_2+k_4-2k_3)\mu_j(r^*)}{\mu_{i+1}(e^*)+\mu_i(e^*)+2\mu_j(r^*)} V \cdot GU \\ &= f(i,j+1)V \cdot GU + \{K_i[e(nT)-\frac{(i+0.5)S}{GE}]+K_p[r(nT)-\frac{(j+0.5)S}{GR}]+\varepsilon\} \end{aligned} \tag{2.13}$$

where

$$K_p = \frac{(2f(i,j+1)-f(i+1,j)-f(i,j))V \cdot GR \cdot GU \cdot S}{2S-2[GR \cdot r(nT)-(j+0.5)S]}$$

$$K_i = \frac{(f(i+1,j+1)-f(i,j+1))V \cdot GE \cdot GU \cdot S}{2S-2[GR \cdot r(nT)-(j+0.5)S]}$$

$\varepsilon = 0$.

Denote $f(i, j+1)V \cdot GU$ as $\Delta u_G(i, j)$ and the rest of the expression as $\Delta u_L(i, j)$. $\Delta u_G(i, j)$ is a global nonlinear controller because it calculates control action with respect to i and j. $\Delta u_L(i, j)$ is a local nonlinear PI-like controller because it calculates control action according to the relative position of the current input state (e(nT), r(nT)) with respect to a dynamically changing point, ((i+0.5)S/GE, (j+0.5)S/GR). K_p and K_i are the proportional-gain and integral-gain. ε is nonzero in some IC regions.

Similar proof can be conducted for the rest of 15 regions. ■

Theorem 2 (General Limit Theorem for Control Rules).

When N approaches ∞,

(2.14)

(1) $\Delta u_L(i, j) = 0$

and

(2) $\Delta u_G(i, j)$ becomes

$$\lim_{i,j,J \to \infty} \frac{f(i,j) \cdot H \cdot GU}{K} \tag{2.15}$$

Proof.

Proof is trivial. ■

Theorem 3.

If linear control rules are used, i.e., if $f(i, j) = -(i + j)$, then

(1) The global nonlinear controller becomes a global two-dimensional multilevel relay

$$\Delta u_G(i,j) = -(i+j+1)\frac{H \cdot GU}{N-1}. \tag{2.16}$$

(2) As N approaches ∞, the global two-dimensional multilevel relay becomes a global linear PI controller:

$$GU \cdot \Delta u(nT) = -(K_i \cdot e(nT) + K_p \cdot r(nT)) \tag{2.17}$$

where

$$K_p = \frac{GR \cdot GU \cdot H}{2L} \tag{2.18}$$

$$K_i = \frac{GE \cdot GU \cdot H}{2L}.$$

Proof.

(1) $K = \text{Max}\{|f(i, j)|\} = 2J = N - 1$. $f(i+1, j) = f(i, j+1) = -(i + j + 1)$. Therefore,

$$\Delta u_G(i,j) = -(i+j+1)\frac{H \cdot GU}{N-1}. \tag{2.19}$$

■

(2) See (Ying, 1991) for proof.

3. Conclusions

With fuzzy control rules being expressed by a function f, the author has been able to analytically reveal the structure of the fuzzy controllers. The structure is the sum of a global nonlinear controller and a local nonlinear PI-like controller.

The work accomplished in this paper furthers understanding on the nature of fuzzy controllers. Fuzzy controllers generally are nonfuzzy nonlinear controllers. Therefore, nonlinear control theory can be utilized to solve fuzzy control problems, such as stability.

References

Ying, H. (Dec. 8, 1987). Fuzzy control theory. Technical Report. Department of Biomedical Engineering, University of Alabama at Birmingham.

Ying, H., W. Siler and J. J. Buckley (1990). Fuzzy control theory: a nonlinear case. Automatica, **26**, 513-520.

Ying, H. (1991). A nonlinear fuzzy controllers with linear control rules is the sum of a Global two-dimensional multilevel relay and a local nonlinear proportional-integral controller. Automatica. to be published.

Fuzzy Control of Small Servo Motors

Ron Maor and Yashvant Jani

Togai InfraLogic, Inc.

To explore the benefits of fuzzy logic and understand the differences between the classical control methods and fuzzy control methods, the Togai InfraLogic applications engineering staff developed and implemented a motor control system for small servo motors. The motor assembly for testing the fuzzy and conventional controllers consist of servo motor RA13M and an encoder with a range of 4096 counts. An interface card was designed and fabricated to interface the motor assembly and encoder to an IBM PC. The fuzzy logic based motor controller was developed using the TILShell and Fuzzy C Development System on an IBM PC. A Proportional-Derivative (PD) type conventional controller was also developed and implemented in the IBM PC to compare the performance with the fuzzy controller. Test cases were defined to include step inputs of 90 and 180 degrees rotation, sine and square wave profiles in 5 to 20 hertz frequency range, as well as ramp inputs. In this paper we describe our approach to develop a fuzzy as well as PD controller, provide details of hardware set-up and test cases, and discuss the performance results. In comparison, the fuzzy logic based controller handles the non-linearities of the motor assembly very well and provides excellent control over a broad range of parameters. Fuzzy technology, as indicated by our results, possesses inherent adaptive features.

A Hierarchical Structure for Representing and Learning Fuzzy Rules

Ronald R. Yager
Machine Intelligence Institute
Iona College
New Rochelle, NY 10801, USA

1. Introduction

In [1] Yager provides an example in which the flat representation [2] of *fuzzy if-then rules* leads to unsatisfactory results. Consider a rule base consisting to two rules

if U is 12 the V is 29 I.
If U is [10-15] the V is [25–30] II.

If U = 12 we would get V is G where G = [25 - 30]. The application of the defuzzification process leads to a selection of V = 27.5. Thus we see that the very specific instruction was not followed.

The problem with the technique used is that the most specific information was swamped by the less specific information. In this paper we shall provide for a new structure for the representation of *fuzzy if-then rules*. The representational form introduced here is called a Hierarchical Prioritized Structure (HPS) representation. Most importantly in addition to overcoming the problem illustrated in the previous example this HPS representation has an inherent capability to emulate the learning of general rules and provides a reasonable accurate cognitive mapping of how human beings store information.

2. Hierarchical Prioritized Structure

Figure 1, shows in a systematic view the of representation of the function V = f(U) by this new HPS representation. The overall function f, relating the input U to the output V, is comprised of the whole collection of subboxes, denoted f_i. Each subbox is a collection of rules relating the system input, U, and the current iteration of the output, V_{i-1}, to a new iteration of the output. The output of the n^{th} subsystem, V_n, becomes the overall output of the system, V. In the HPS the higher priority boxes, *for i < j we say that fi has a higher priority than fj*, would have less general information, consist of rules with more specific antecedents then those of lower priority. As we envision this system working an input value for U is provided, if it matches one or more of the rules in the first (highest priority) level then it doesn't bother to fire any of the less specific rules in the lower priority levels.

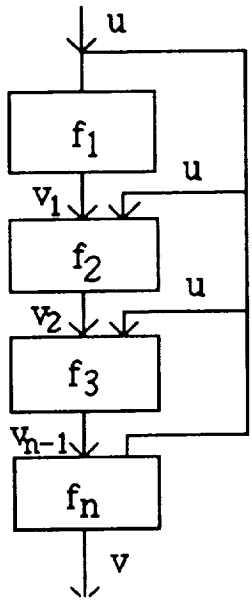


Figure 1 Hierarchical Prioritized Structure

In the following we describe the formal operation of this HPS. As we indicated V_j denotes

the output of the j^{th} level. We shall assume $V_0 = \Phi$. In the HPS we shall use the variable $\hat{V}_j$ to indicate the **maximum membership grade** associated with the output of the j^{th} level, V_j.

In the HPS each f_j (accept for the lowest level, j = n) is a collection of n_j rules

When U is A_{ji} is certain and $\hat{V}_{j-1}$ is low then V_j is B_{ji} **I**

The representation and aggregation of rules at each level is of the standard Mamdani type[2], disjunction of the individual rules. If $\widetilde{B}_j$ is the value obtained from the aggregation of the outputs of the collection of individual rules in I then the output of this subbox is

$$V_j = V_{j-1} \cup \widetilde{B}_j.$$

In **I** a rule fires if we are certain that the input U lies in A_{ji} and $\hat{V}_{j-1}$ is low. Since $\hat{V}_{j-1}$ is the maximum membership grade of V_{j-1} it can be seen as a measure of how much matching we have up to this point. Essentially this term is saying that if the higher priority rules are relevant, $\hat{V}_{j-1}$ is not low, then don't bother using this information. On the other hand if the higher priority rules are not relevant, not to much matching $\hat{V}_{j-1}$ is low, then try using this information.

The representation of the box f_n is a collection of rules

When U is A_{ni} and $\hat{V}_{n-1}$ is low then V is B_{ni} **II**

plus the rule $V = V_n = V_{n-1} \cup \widetilde{B}_n$. The notable difference between the lowest priority box and the other ones is that the antecedent regarding U is certainly quality in the higher boxes. The need for this becomes apparent when the input is not a singleton.

In the HPS structure $\hat{V}_{j-1}$ is the highest membership grade in V_{j-1} and as such the term $\hat{V}_{j-1}$ *is low* is used to measure the degree to which the higher prioritized information have matched the input data. We note that low is a fuzzy subset on the unit interval. One definition for low [1] is

$$\text{low}(x) = 1 - x.$$

In [1] Yager looks at the formal operation of this kind of HPS we shall present the results obtained in [1]. We shall denote $\Im_{ij}$ as the degree of firing (or relevancy) of A_{ij} under the input, if the input is U = x* then $\Im_{ij} = A_{ij}(x^*)$. We shall denote $g_i = \text{Max}_y\, G_i(y) = \text{Poss}[G_i]$. We let $T_i = \bigcup_{j=1}^{n_i} \Im_{ij} \wedge B_{ij}$, the aggregation of the rules in the ith level for input U, it is essentially the contribution of the ith subsystem using the Mamdani type reasoning.. We shall let G_i be the output of the ith subbox, that is $V_i = G_i$. In [1] it is shown that

$$G_i(y) = (T_i(y) \wedge (1 - g_{i-1})) \vee G_{i-1}(y). \qquad \textbf{III}$$

We notice that the term $(1 - g_{i-1})$ bounds the allowable contribution of the i^{th} subsystem to the overall output. We see that as we get at least one element y to be, a good answer (an element in G_{i-1}) we limit the contribution of the lower priority subsystems. It is this characteristic of a kind of saturation along with the prioritization that allows us to avoid the problem described earlier.

In the following we suggest a modification of the above that leads to a more suitable formulation to the aggregation between the levels of the HPS [1]. We can replace $\wedge$ by another t-norm operator product * and replace $\vee$ by another t-conorm, bounded sum, $a \boxed{+} b = \text{Min}(1, a+b)$[3]. Thus we get

$$G_i(y) = T_i(y) * (1 - g_{i-1}) \boxed{+} G_{i-1}(y).$$

However we note that since $g_{i-1} = \text{Max}_y G_{i-1}$ then $T_i(y) * (1 - g_{i-1}) \leq G_{i-1}(y)$ hence $T_i(y) * (1 - g_{i-1}) + G_{i-1}(y) \leq 1$ thus we can replace $\boxed{+}$ by +. This gives us the formulation

$$G_i(y) = T_i(y) * (1 - g_{i-1}) + G_{i-1}(y) \qquad \textbf{(IV)}$$

$$G_i(y) = T_i(y) * (1 - \text{Poss}[G_{i-1}]) + G_{i-1}(y)$$

What is happening in this structure is that as long as we have not found one y with

membership grade 1 in G_{i-1}, $Poss[G_{i-1}] \neq 1$, we add some of the output of the current subbox to what we already have. Each element y, gets $1 - Poss[G_{i-1}]$ portion of the contribution at that level, $T_i(y)$

We should point out that the aggregation performed in the hierarchical structure, whether we use III or IV, is not a pointwise operation. This means that the value of $G_i(y)$ doesn't only depend on the membership grade of y in G_{i-1} and T_i but on membership grades at other points. In particular through the term $\bar{g}_{i-1} = 1 - Max_y\ G_i(y)$ it depends upon the membership grade of all elements from Y in Gi–1.

We should note that implicit in this structure is a new kind of aggregation. Assume A and B are two fuzzy subsets we define the combination of these sets as the fuzzy subset D, denoted $D = \gamma(A, B)$ where

$$D(x) = (1 - Poss(A)) * B(x) + A(x).$$

3. Representation and Operation of the HPS

In the previous part we have described the formal mechanism used for the reasoning and aggregation process in the HPS. While the formal properties of the new aggregation structure are important a key to the usefulness of the HPS in fuzzy modeling is the semantics used in the representation of the information via this structure.

In constructing an HPS representation to model a system we envision that the knowledge of the relationship contained in the HPS structure be stored in the following manner. At the highest level of priority, i = 1, we would have the most specific precise knowledge. In particular we would have point to point relationships,

When U is 3 then V is 7
When U is 9 then V is 13

This would be information we know with the greatest certainty.

At the next level of priority the specificity of the antecedent linguistic variables, the A_{2j}'s, would decrease. Thus the second level would contain slightly more general knowledge.

Essentially what we envision is that at the highest level we have specific point information. The next level encompass these points and in addition provides a more general and perhaps fuzzy knowledge. We note that the lowest most level can be used to tell us what to do if we have no knowledge up to this point. In some sense the lowest level is a default value.

Example: Assume we are using an HPS representation to model a function V = f(U), where the base set for U is [0, 100]. A typical HPS representation could be as follows.

LEVEL #1

R_{11} When U is 5 then V is 13
R_{12} When U is 75 then V is 180
R_{13} When U is 85 then V is 100

LEVEL #2

R_{21} When U is "about 10" then V is "about 20"
R_{22} When U is "about 30" then V is "about 50"
R_{23} When U is "about 60" then V is "about 90"
R_{24} When U is "about 80" then V is "about 120"
R_{25} When U is "about 100" then V is "about 150"
(we assume triangular fuzzy subsets)

LEVEL #3

R_{31} When U is "low then V is "about 40"
R_{32} When U is ":meet" then V is "about 85"
R_{31} When U is "high" then V is "about 130"

LEVEL #4

R_{41} U is anything the V is 2u.

Having defined our knowledge base we now look at the performance of this system for various inputs;

Case 1: U = 75. At level one we get $T_1 = \left\{\frac{1}{180}\right\}$ hence since

$$G_1(y) = \bar{g}_0 * T_1(y) + G_0(y).$$

Since $G_0 = \Phi$ then $g_1 = 1$ which give us $G_1 = T_1 = \left\{\frac{1}{180}\right\}$. We now see that $\bar{g}_1 = 0$ and hence no other rules will fire lower in the hierarchy. This system provides as its output for U = 75 that V is 180.

Case 2: U = 80. At level no rules fire, $\Im_{ij} = 0$ for all j. Thus $T_1 = \Phi$ hence

$$G_1 = \bar{g}_0 * T_1 + G_0 = \Phi$$

and therefore $\bar{g}_1 = 1$. At level two

$$G_2 = \bar{g}_1 * T_2 + \Phi = T_2.$$

For U = 80 we assume that R_{24} fires completely, $\Im_{24} = 1$ and that all other rules don't fire, $\Im_{2j} = 0$, for $j \neq 2$. Thus T_2 = "about 120" and G_2 = "about 120". Since $g_2 = 1$ then $\bar{g}_2 = 0$ and no rules at lower priority will fire thus G_2, "about 120", is the output of the system for U = 80.

Case 3: U = 20. No rule at level one will fire, hence $G_1 = G_0 = \Phi$. At level two we shall assume that R_{21} fires to degree .3 and R_{22} also fires to degree .3. Thus

$$T_2 = .3 \wedge B_1 \cup .3 \wedge B_2 = .3 \wedge (B_1 \cup B_2)$$

$$T_2(y) = .3 \wedge (B_1(y) \vee B_2(y)).$$

We note B_1 and B_2 are "about 20" and "about 30" respectively. Hence

$$G_2(y) = (1 - g_1) * T_2(y) + G_1(y) = T_2(y)$$

At level three R_{31} fires to degree 1 while R_{31} and R_{32} don't fire at all. Hence

$$T_3 = \text{"about 40"}$$

Since Max $[G_2] = .3$ thus $1 - g_2 = .7$ and therefore

$$G_3(y) = .7 * T_3(y) + G_2(y)$$

Since Max $T_3(y) = 1$ we see that the process stops here and G_3 is the output of the system.

What we see with this HPS representation is that we have our most general rule stored at the *lowest* level of priority and we store exceptions to this rule at higher levels of priority. In some cases the exceptions to general rules may themselves be rules, we would then store exceptions to these rules at still higher levels of priority. As the previous example illustrates in the HPS system for a given input we first look to see if the input is an exception, that is what we are essentially doing by looking at the high priority levels.

4.Learning in the HPS

The HPS representation is a formulation that has an inherent structure for a natural human like learning mechanism. We shall briefly describe the type of learning that is associated with this structure.

Information comes into the system in terms of point by point knowledge, data pairs between input and output. We store these points at the highest level of priority. Each input/output pair corresponds to a rule at the highest level. If enough of these points cluster in a neighborhood in the input/output space we can replace these points by a general rule (see figure 2).

Thus from the dots, input/output pairs, we get a relationship that says *if U is in A then V is B*. We can now forget about the dots and only save the new relationship. We save this at the next lowest priority in the system, in subbox 2.

We note that the introduction of the rule essentially extends the information contained in the dots by now providing information about spaces between the dots. We can also save storage because we have eliminated many dots and replaced them by one circle. One downside to this formulation is that in generalizing we have lost some of the specificity carried by the dots.

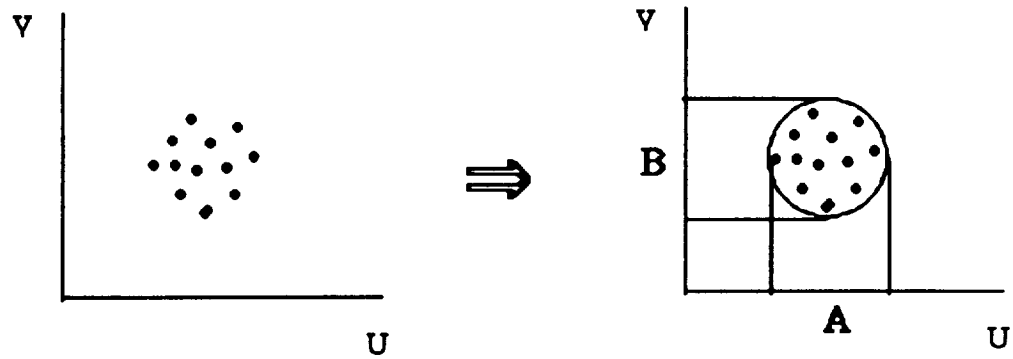


Figure 2 Formulation of Rules for Input/Output Pairs

It may occur that there are some notable exceptions to this new general rule. We are able to capture this exception by storing them as high level points.

We further note that new information enters the system in terms of points. Thus we see that the points are either new information or exceptions to more general rules. Thus specific information enters as points it filters its way up the system in rules.

We see that next that it may be possible for a group of these second level rules to be clustered to form new rules at the third level.

In figure #3 the large bold circle is seen as a rule which encompasses the higher level rules to provide a more general rule. The necessity to keep these more specific rules, thus in level 2, depends upon how good the less specific rule captures the situation.

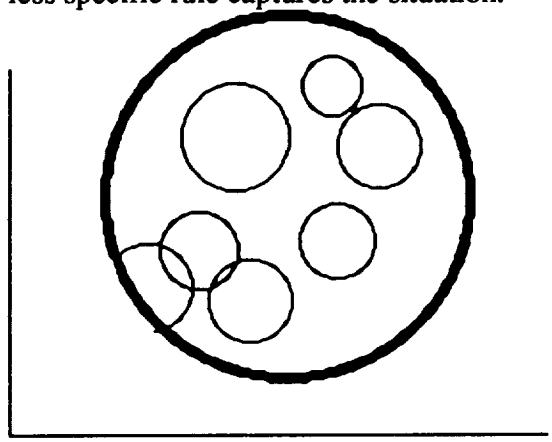


Figure 3 Aggregation of Rules into More General Rules

5. References

[1]. Yager, R. R., "On structures for fuzzy modeling and control," Technical Report MII#1213, Machine Intelligence Institute, Iona College, 1992.

[2]. Mamdani, E. H. and Assilian, S., "An experiment in linguistic synthesis with a fuzzy logic controller," Int. J. of Man-Machine Studies 7, 1-13, 1975.

[3]. Dubois, D. and Prade, H., "A review of fuzzy sets aggregation connectives," Information Sciences 36, 85 - 121, 1985.

CERTAIN & POSSIBLE RULES FOR DECISION MAKING USING ROUGH SET THEORY EXTENDED TO FUZZY SETS

André de Korvin

Department of Applied Mathematical Sciences
University of Houston-Downtown

Margaret F. Shipley

Department of Business Management
University of Houston-Downtown
One Main Street
Houston, Texas 77002

(713) 221- 8571
FAX (713) 221- 8632

1. Introduction

Uncertainty may be caused by the ambiguity in the terms used to describe a specific situation. It may also be caused by skepticism of rules used to describe a course of action or by missing and/or erroneous data. [For a small sample of work done in the area, the reader is referred to (Arciszewski & Ziarko 1986), (Bobrow, et.al. 1986), (Wiederhold, et. al. 1986), (Yager 1984), and (Zadeh 1983).]

To deal with uncertainty, techniques other than classical logic need to be developed. Although, statistics may be the best tool available for handling likelihood, it is not always adequate for dealing with knowledge acquisition under uncertainty. [We refer the reader to Mamdani, et. al. (1985) for a study of the limitations of traditional statistical methods.]

Inadequacies caused by estimating probabilities in statistical processes can be alleviated through use of the Dempster-Shafer theory of evidence. [For a sample of works using the Dempster-Shafer theory see (Shafer 1976), (de Korvin, et. al. 1990), (Kleyle & de Korvin 1989), (Strat 1990), and (Yager).] Fuzzy set theory is another tool used to deal with uncertainty where ambiguous terms are present. [Articles in (Zadeh 1979, 1981 & 1983) illustrate the numerous works carried out in fuzzy sets.] Other methods include rough sets, the theory of endorsements and nonmonotonic logic. [The work on rough sets is illustrated in (Fibak, et. al. 1986),

(Grzymala-Busse 1988), and (Mrozek 1985 & 1987). Also, see (Mrozek 1985) and (Pawlak 1982) for the application of rough sets to medicine and (Arciszewski & Ziarko 1986) and (Pawlak 1981) for applications to industry.]

J. Grzymala-Busse (1988) has defined the concept of lower and upper approximation of a (crisp) set and has used that concept to extract rules from a set of examples. We will define the fuzzy analogs of lower and upper approximations and use these to obtain certain and possible rules from a set of examples where the data is fuzzy. Central to these concepts will be the idea of the degree to which a fuzzy set A is contained in another fuzzy set B, and the degree of intersection of set A with set B. These concepts will also give meaning to the statement; A implies B. The two meanings will be: 1) if x is certainly in A then it is certainly in B, and 2) if x is possibly in A then it is possibly in B. Next, classification will be looked at and it will be shown that if a classification is well externally definable then it is well internally definable, and if it is poorly externally definable then it is poorly internally definable, thus generalizing a result of Grzymala-Busse (1988). Finally, some ideas of how to define consensus and group opinions to form clusters of rules will be given.

2. Results

We now recall some basic definitions such as lower and

upper approximations and the concept of an information system.

Let U be the universe. Let R be an equivalence relation on U. Let X be any subset of U. If [x] denotes the equivalence class of x relative to R, then we define

$\underline{R}(X) = \{x \in U/[x] \subset X\}$ and

$\overline{R}(X) = \{x \in U/[x] \cap X \neq \emptyset\}$.

$\underline{R}(X)$ is called the lower approximation of X and $\overline{R}(X)$ is called an upper approximation of X. Then $\underline{R}(X) \subset X \subset \overline{R}(X)$. If $\underline{R}(X) = X = \overline{R}(X)$, then X is called definable.

An information system is a quadruple (U,Q,V,τ) where U is the universe and Q is a subset of $C \cup D$ where $C \cap D = \emptyset$. The set C is called the set of conditions; D is called the set of decisions. We assume here that $Q = C$. The set V stands for value and τ is a function from UxQ into V where $\tau(u,q)$ denotes the value of attribute q for element u. The set C induces naturally an equivalence on U by partitioning U into sets over which all attributes are constant. The set X is called roughly C-definable if

$\underline{R}(X) \neq \emptyset$ and $\overline{R}(X) \neq U$.

It will be called internally C-undefinable if

$\underline{R}(X) = \emptyset$ and $\overline{R}(X) \neq U$.

It will be called externally C-undefinable if

$\underline{R}(X) \neq \emptyset$ and $\overline{R}(X) = U$.

Fuzzy sets defined

Next, we define two functions on pairs of fuzzy sets that will be of importance in the present work.

$$I(A \subset B) = \inf_x \text{Max } \{1 - A(x), B(x)\} \quad (1)$$

$$J(A \# B) = \text{Max}_x \text{ Min } (A(x), B(x)). \quad (2)$$

Here A and B denote fuzzy subsets of the same universe. The function $I(A \subset B)$ measures the degree to which A is included in B and $J(A \# B)$ measures the degree to which A intersects B. It is important to note that for the crisp case, $I(A \subset B) = 1$ iff $A \subset B$ and is 0 otherwise. Similarly, $J(A \# B) = 1$ iff $A \cap B \neq \emptyset$.

The goal is to define the fuzzy terms involved in the decision as a function of the terms used in the conditions. This is accomplished as a function of how much the decision follows the conditions. Let $\{B_i\}$ be a finite family of fuzzy sets. Let A be a fuzzy set. By a lower approximation of A through $\{B_i\}$, we mean the fuzzy set

$$\underline{R}(A) = \bigcup_i I(B_i \subset A) B_i \quad (6)$$

The decision making process may be simplified by disregarding all sets B, if $I(B_i \subset A)$ is less than some threshold α. Then,

$$\underline{R}(A)_\alpha = \bigcup_i I(B_i \subset A) B_i \quad (7)$$

over all B_i for which $I(B_i \subset A) \geq \alpha$.

Similarly, we can define the upper approximation of A through $\{B_i\}$ as

$$\overline{R}(A)_\alpha = \bigcup_i J(B_i \# A) B_i \quad (8)$$

over all B_i for which $J(B_i \# A) \geq \alpha$.

The operators I and J will yield two possible sets of rules: the certain rules and the possible rules. It is straightforward to see that if $\{B_i\}$ are crisp equivalency

classes we get the lower and upper approximations as defined by Grzymala-Busse (1988).

Determining Fuzzy Rules

We now show how rules can be obtained from the raw data given in Table 1 after converting this data according to the professor's evaluation of the performance of the students, relative to exams high, exams low, project high, project low, and his belief with respect to each student getting an A. (See Table 2 for the converted data.)

Table 1: Production/Operations Management Grades

Student	Exams (2)	Project (Written & Oral)	Course Grade
1	75	85	75.36
2	94	87	89.53
3	88	89.3	89.93
4	79.5	95	78.06
5	85	97	90.85
6	56.5	88.6	60.89
7	65	91.6	76.15
8	49	76.7	59.22
9	63.5	89.1	69.99
10	57	76.9	55.77
11	70	98	80.3
12	93	88	90.1

It can be observed that none of the course grades was a strong predictor of "success". In other words, the course grades of 90 or slightly better than 90 as a "quality" measure of the final product did not allow the professor strong belief in the awarding of an "A" to the student. The professor's belief in these grades being the best in the class and therefore deserving of an "A" grade was approximately .67. The

belief in the lower scores is scaled downward from .67 to .41 (the latter representing belief that 55.77 will be the top score in the class.)

The professor recognized the high exam scores of 94 and 93, with belief of .99/EH and .98/EH, respectively (EH: Exams High). The low exam score of .49 was designated .92/EL (EL: Exams Low) by the professor. Since all project grades were relatively close and relatively high, the professor saw little differentiation between the "top" score and the other scores. The "top" project score is .54 high and .46 low. (.54/PH and .46/PL, respectively) This contrasts with the worse project score being .43/EH and .59/EL, where .59 is the highest belief that a project grade is a "low" score. This approach was considered to be consistent since although exam grades varied from 49 to 94, no project grade was below a 76.7. It was felt that keeping the project grades from being too strongly biased toward "high" would prevent the decision rules from being overly biased toward high project grades. Enough differentiation was considered to allow the rough set formulation to consider both attributes in the decision rules for awarding a "top" score of "A" to a student. Each student's scores were translated into belief with respect to EH, EL, PH, PL and "A".

For our example of twelve POM students, $x_1, x_2, \ldots, x_{12}$, we let EH:exams high PH:project high

EL:exams low PL:project low "A": Top Grade

Thus, for the first student, x_1, the belief that the exams were high is .79/EH, and that the exams were low is .60/EL; that the project grade was high is .47/PH and that it was low is .53/PL. The strength of belief for an A is .56/"A". In addition, EH may be viewed as a fuzzy set of students, such that EH = $.79/x_1 + .99/x_2 + \ldots + .98/x_{12}$, where x_2 is an excellent example of EH (.99) while x_8 is not such a good example (.52). (See Table 2 below for all the professor's evaluative scores.)

Table 2: Professor's Evaluative Scores

Student	EH	EL	PH	PL	"A"
1	.79	.60	.47	.53	.56
2	.99	.48	.48	.52	.66
3	.93	.51	.50	.50	.67
4	.84	.57	.53	.47	.58
5	.89	.53	.54	.46	.67
6	.58	.81	.49	.51	.45
7	.68	.69	.51	.49	.56
8	.52	.92	.43	.58	.44
9	.67	.71	.50	.51	.52
10	.60	.79	.43	.59	.41
11	.74	.64	.54	.46	.59
12	.98	.48	.49	.51	.67

Using our rough set theory formulas as they have been developed for fuzzy systems of attributes and decisions, we compute:

$I(EH \subset \text{"A"}) = .41$ $\quad$ $I(EH \cap PH \subset \text{"A"}) = .51$

$I(EL \subset \text{"A"}) = .41$ $\quad$ $I(EH \cap PL \subset \text{"A"}) = .42$

$I(PH \subset \text{"A"}) = .51$ $\quad$ $I(EL \cap PH \subset \text{"A"}) = .51$

$I(PL \subset \text{"A"}) = .42$ $\quad$ $I(EL \cap PL \subset \text{"A"}) = .42$

with a lower approximation for $\alpha = .50$ defined by:

$\underline{R}$ =.51 PH ∪ .51 (EH ∩ PH).

The extracted rules would imply that high project scores and high exam scores both impact a high course grade with certainty .51.

Possibility rules can be determined by computing:

J(EH #"A") = .67 J(EH ∩ PH # "A") = .54

J(EL # A") = .59 J(EH ∩ PL # "A") = .53

J(PH #"A") = .54 J(EL ∩ PH # "A") = .54

J(PL #"A") = .53 J(EL ∩ PL # "A") = .53

with an upper approximation at α = .60 defined as:

$\overline{R}$ = .67 EH.

Thus, we can see that the factors dictating the "best" in the class are:

1) If project grades are high, an "A" score will be attained. (Certainty = .51)
2) If project grades and exam grades are high, an "A" score will be attained. (Certainty = .51)
3) If exam grades are high, an "A" score will be attained. (Possibility = .67)

Indeed, these rules reflect the fact that exam grades are more heavily weighted than the project grade toward determining the final course grade. Additionally, these two grades comprise the majority of the weighted scores from which the course grade is calculated.

Belief & Possibility

We can use the functions I and J to determine two

meanings of A implies B. The belief that if x is certainly in A then it is certainly in B is given by:

$$I[\ \underline{R}\ (A) \subset\ \underline{R}\ (B)\ ] \tag{9}$$

and the belief that if x is possibly in A then it is possibly in B can be defined by:

$$J[\ \overline{R}\ (A)\ \#\ \overline{R}\ (B)\ ] \tag{10}$$

This interpretation follows from the fact that $\underline{R}(A)$ are objects certainly in A and $\overline{R}(A)$ are objects possibly in A. We now turn to the study of classifications.

Classifications

The study of classifications is of great interest because in learning from examples, the rules are derived from classifications generated by simple decisions. In this section, we turn our attention to classifications. Of course, the traditional meaning is to partition. In our setting, we have ill-defined boundaries, so we need to relax the concept of partitions by requiring that the sets not overlap too much.

As earlier, consider a finite family of fuzzy sets, $\{B_i\}$. Let π denote a finite family of fuzzy sets

$$\pi = \{A_1, A_2, \ldots, A_n\}$$

We define

$$\underline{P}\pi_\alpha = \{\ \underline{R}(A_1)_\alpha, \ldots, \underline{R}(A_n)_\alpha\},$$

$$\overline{P}\pi_\alpha = \{\ \overline{R}(A_1)_\alpha, \ldots, \overline{R}(A_n)_\alpha\}$$

where the lower and upper α-approximations are generated by the finite sequence $\{B_i\}$.

We can develop the following relationship:

$d^*[A = B] = \text{Min}\ \{\ I\ (A \subset B),\ I(\ B \subset A)\}$

using the following definitions:

$d^*[\underline{P}\pi_\alpha = \pi] = \underset{k}{\text{Min}}\ \{d^*[\underline{R}(A_k)_\alpha = A_k]\}$

$d^*[\overline{P}\pi_\alpha = \pi] = \underset{k}{\text{Min}}\{d^*[\overline{R}(A_k)_\alpha = A_k]\}$

π will be called $\{B_i\}$ definable to the degree β with threshold α if

$\text{Min}\ \{\ d^*[\underline{P}\pi_\alpha = \pi],\ d^*[\overline{P}\pi_\alpha = \pi]\} \geq \beta.$

If we define

$d^*[\underline{P}\pi_\alpha = \overline{P}\pi_\alpha] = \underset{k}{\text{Min}}\ \{\ d^*[\underline{R}(A_k)_\alpha = \overline{R}(A_k)_\alpha]\},$

it can be shown that if $\beta \geq \frac{1}{2}$, then

$d^*[\underline{P}\pi_\alpha = \pi] \geq \beta$ and $d^*[\overline{P}\pi_\alpha = \pi] \geq \beta$ imply that

$d^*[\underline{P}\pi_\alpha = \overline{P}\pi_\alpha] \geq \beta.$

Recall that the following result is shown in information systems. For classifications, if $\overline{P}A_k$ is the universal set for each k, then $\underline{P}A_k$ is empty for each k. Also, if $\underline{P}A_k$ is nonempty for each k, the $\overline{P}A_k$ is not the universal set for any value of k. We would like to get the analog of this by showing if $\underline{R}(A_k)_\alpha$ "has some substance" for some k, then $\overline{R}(A_j)_\alpha$ for $j \neq k$ is "not too large", and if $\overline{R}(A_k)_\alpha$ is "fairly substantial", $\underline{R}(A_j)_\alpha$ for $j \neq k$ cannot be "too large". In this sense, the results of Grzymala-Busse (1988) will be generalized.

We would like $\{A_k\}$ and $\{B_i\}$ to somewhat approximate a partition. We define the following two conditions:

(*) For every $0 < \epsilon < 1$, there exists $0 < \delta < 1$ such that if $B_i(x_0) > \epsilon$, then $B_\ell(x_0) < 1 - \delta$ for $\ell \neq i$.

(**) For every pair j,k with $j \neq k$ and all x, $A_k(x) + A_j(x) \leq 1$.

Conditions (*) and (**) both express that the overlap is not too large and obviously hold for partitions. We note that if (**) holds for $\{B_i\}$ then it implies (*). Indeed, in this case we pick $\delta = \epsilon$. Thus, the results that follow may be shown assuming condition (**) for $\{B_i\}$ and $\{A_k\}$.

We first show that under conditions (*) and (**), whenever $\underline{R}(A_k)_\alpha$ is bounded away from 0, then $\overline{R}(A_j)_\alpha$ for $j \neq k$ is bounded away from 1. Suppose $\underline{R}(A_k)_\alpha(x_0) > \epsilon$, then for some i, $I(B_i \subset A_k) > \epsilon$ and $B_i(x_0) > \epsilon$, so for $\ell \neq i$ from condition (*), we have $B_\ell(x_0) < 1 - \delta$. For any $\ell \neq i$ we have $J(B_\ell \# A_j)B_\ell(x_0) < 1 - \delta$. Now

$$J(B_i \# A_j) = 1 - I(B_i \subset \neg A_j);$$

$$I(B_i \subset A_k) = \min_x \max \{1-B_i(x), A_k(x)\};$$

$$I(B_i \subset \neg A_j) = \min_x \max \{1-B_i(x), 1-A_j(x)\}.$$

Condition (**) implies $I(B_i \subset A_k) \leq I(B_i \subset \neg A_j)$ for all $j \neq k$. From the above it follows that $J(B_i \# A_j) < 1 - \epsilon$. Thus,

$$\overline{R}(A_j)_\alpha(x_0) < \max \{ 1- \epsilon, 1- \delta\}.$$

We now show a rough converse to the above. If $\overline{R}(A_k)$ is bounded away from 0, then for $j \neq k$, $\underline{R}(A_j)_\alpha$ is bounded away from 1. Suppose $\overline{R}(A_k)_\alpha(x_0) > 1 - \epsilon$ for some k, then

$$J(B_{i_0} \# A_k)B_{i_0}(x_0) > 1 - \epsilon \text{ for some } i_0.$$

Pick $j \neq k$. Then

$$I(B_{i_0} \subset A_j) = 1 - J(B_{i_0} \# \neg A_j).$$

Now, $J(B_{i_0} \# \neg A_j) = \max_x \min \{B_{i_0}(x), 1 - A_j(x)\};$

$$J(B_{i_0} \# A_k) = \max_x \min \{B_{i_0}(x), A_k(x)\}.$$

By (**) it follows that $J(B_{i_0} \# \neg A_j) \geq J(B_{i_0} \# A_k)$.

From above, $I(B_{i_0} \subset A_j) \leq 1 - J(B_{i_0} \# A_k) \leq \epsilon$.

Since $B_{i_0}(x_0) > 1 - \epsilon$, by (*), $B_i(x_0) < \theta$ for $i \neq i_0$ where $0 < \theta < 1$.

Therefore, $\underline{R}(A_j)_\alpha(x_0) \leq \text{Max} \{ \epsilon, \theta \}$.

Consensus

We can define consensus between two rows of a table by

Consensus $[\text{Row}_i, \text{Row}_j] = \text{Min} \{ I[\text{Row}_i \subset \text{Row}_j], I[\text{Row}_j \subset \text{Row}_i] \}$

Here, Row_i and Row_j are considered to be fuzzy subsets of the set of all attributes and decisions. If γ is some predetermined threshold, we pick some x_1 and then all x_j for which Consensus $[\text{Row}_1, \text{Row}_j] \geq \gamma$. If any of the x's are left over, we start again with the first x available. We thus get fuzzy sets $S_1, S_2, \ldots, S_\ell$ where $\mu_{S_i}(\ell_i) = 1$ for some ℓ_i (which we might call the leader of S_i) and $\mu_{S_i}(x) = \text{Consensus}(l_i, x)$ provided $\mu_{S_i}(x)$ exceeds γ. Within each S_i we then can recompute the symptoms/decisions for x_j taking $\mu_{S_i}(x_j)$ into account If $1 \leq i \leq \ell$, then we have ℓ (aggregated) decisions and using fuzzy cardinality we can compute the "firing strength" of each block of rules. This approach has the advantage of taking consensus of opinions into consideration in the decision. The detailed methodology will be discussed in a later paper.

REFERENCES

Arciszewski, T. and Ziarko, W. 1986. "Adaptive expert system for preliminary engineering design," Proceedings 6th International Workshop on Expert Systems and their Applications. Avignon, France, 1, 696-712 .

Bobrow, D.G., Mittal, S. and Stefik, M.J. 1986. "Expert systems perils and promises" Communications of the ACM, 29, 880-894.

Cheeseman, P. 1986. "Induction of models under uncertainty," Proceedings of ACM SIGART International Symposium on

Methodologies for Intelligent Systems, Knoxville, Tennessee, 130-144.

Fibak, J., Slowinski, K. and Slowinski, R. 1986. "The application of rough set theory to the verification of indications for treatment of duodenal ulcers by HSV," Proceedings 6th International Workshop on Expert Systems and their Applications, Avignon, France, 1, 587-594.

Grzymala-Busse, J.W. 1988. "Knowledge acquisition under uncertainty: a rough set approach," Journal of Intelligent and Robotic Systems 1, 3-16.

Kleyle, R. and de Korvin, A. 1989. "A unified model for data acquisition and decision making," The Journal of the American Society for Information Science 15, 149-161.

de Korvin, A., Kleyle, R. and Lea, R. 1990. "An evidential approach to problem solving when a large number of knowledge systems are available," The International Journal of Intelligent Systems, 5, 293-306.

Mamdani, A., Efstathiou, J. and Pang, D. 1985. "Inference under uncertain expert systems 85," Proceedings Fifth Technical Conference British Computer Society, Specialist Group on Expert Systems, 181-194.

Mrozek, A. "Information systems and control algorithms", 1985. Bulletin Polish Academy of Science, Technical Science, 33, 195-204.

Mrozek, A. 1987. "Rough sets and some aspects of expert systems realization," Proceedings 7th International Workshop on Expert Systems and their Applications, Avignon, France, 597-611.

Pawlak, Z. 1981. "Rough sets. Basic Notions," Institute Computer Science, Polish Academy of Science Report No. 431, Warsaw.

Pawlak, Z. 1981. "Classification of objects by means of attributes," Institute Computer Science Polish Academy Science Report No. 429, Warsaw.

Pawlak, Z. 1982. "Rough sets," International Journal of Information Computer Science ,11, 341-356.

Pawlak, Z. 1983. "Rough classifications," International Journal of Man-Machine Studies, 20, 469-483.

Pawlak, Z. 1985. "Rough sets and fuzzy sets," Fuzzy Sets and Systems, 17, 99-102.

Shafer, G. 1976. Mathematical Theory of Evidence, Princeton University Press.

Strat, T.M. 1990. "Decision analysis using belief functions," International Journal of Approximate Reasoning, 4, 391-417.

Wiederhold, G.C., Walker, M., Blum, R., and Downs, S. 1986. "Acquisition of knowledge from data," Proceedings ACM SIGART International Symposium on Methodologies for Intelligent Systems, Knoxville, Tennessee, 78-84.

Yager, R.R. 1984. "Approximate reasoning as a basis for rule based expert systems," IEEE Transactions on Systems, Man and Cybernetics, 14, 636-643.

Yager, R.R. "Decision making under Dempster-Shafer uncertainties," Iona College, Machine Intelligence Institute Tech. Report MII-915.

Zadeh, L.A. 1983. "The rule of fuzzy logic in the management of uncertainty in expert systems," Fuzzy Sets and Systems 11, 119-227.

Zadeh, L.A. 1979. "Fuzzy sets and information granularity," Advances in Fuzzy Set Theory and Applications, 3-18.

Zadeh, L.A. 1981. "Possibility theory and soft data analysis," Mathematical Frontiers of the Social and Policy Sciences. Eds. L. Cobb and R.M. Thrall, 69-129. Westview Press, Boulder, Colorado.

On Structuring the Rules of a Fuzzy Controller

(summary)

Jun Zhou, G. V. S. Raju

Division of Engineering

The University of Texas at San Antonio

San Antonio, TX 78249-0665

Since the pioneering work of Zadeh[1] and Mamdani and Assilian[2], fuzzy logic control has emerged as one of the most active and fruitful research areas[3][4]. The applications of fuzzy logic control can be found in many fields such as control of steam generators, automatic train operation systems, elevator control, nuclear reactor control, automobile transmission control , etc.

In most of existing fuzzy rule-based controllers, the rules are based upon the error and the change in error, where the error is defined as the difference between the desired output and actual output. However, in a large-scale system, the signals error and change in error only provide a limited amount of information about system status. Therefore, the performance of the controller will be limited, since only a fraction of the feedback information will be available to the controller. To avoid this limitation, the fuzzy rules need to be based upon more system variables. It is well known that the total number of rules in a complete rule set is a exponential function of the system variables on which the rules are based. As such when more system variables are used, the number of the rules will increase exponentially. This will make the fuzzy rule-based controller more complex as well as expensive to realize.

To make the problem manageable, the concept of a 'hierarchical fuzzy rule set' was introduced in reference[5]. In a hierarchically structured rule base, the number of rules

increases linearly (not exponentially) with the number of system variables. This makes it possible to apply a fuzzy rule-based controller to large-scale systems.

In this paper, two new structures of hierarchical fuzzy rule-based controller are proposed to reduce the number of rules in a complete rule set of a controller. In one approach, the overall system is split into sub-systems which are treated independently in parallel. A coordinator is then used to take into account the interactions. This is done via an iterating information exchange between the lower level and the coordinator level. Figure 1 schematically shows the main idea. From the point of view of information used, this structure is very similar to central structure in that the coordinator can have at least in principle, all the information that the local controllers have.

A more general structure of this approach is shown in Fig. 2, where more coordinate levels are introduced. By using this hierarchical structure, the theoretical minimum total number of rules will be a linear function of the system variables. The actual total is dependent upon the number of system variables used in each local controller's rule sets and coordinator's rule sets. Specifically, if we denote N as the total number of rules, then

$$N = \sum_{i=1}^{i=N_{1i}} m^{n_{1i}} + \sum_{i=1}^{i=N_{2i}} m^{n_{2i}} + \sum_{i=1}^{i=N_{3i}} m^{n_{3i}} + \ldots$$

$$= \sum_{j=1}^{j=L} \sum_{i=1}^{i=N_{ji}} m^{n_{ji}}$$

where N_{ji} is the number of local controllers or coordinators in the jth level, n is the number of variables used in the ith local controller or coordinator in the jth level, and m is the number of the linguistic fuzzy variables used in each local controller or coordinator.

One important advantage of this approach over that in reference[5] is that all the rule

sets in the same level can be fired simultaneously. As such, this approach will be more suitable for parallel computing. However, using the structure in reference[5], additional system variables can be easily included in the fuzzy rule set without affecting other rules. A more versatile hierarchical structure, combining the hierarchical structure proposed in Fig. 2 and that in reference[5], will be presented in the paper. This versatile structure will have the advantages of all the structures discussed earlier.

References

[1] L. A. Zadeh, "Fuzzy algorithm," Information and Control, Vol. 12, pp. 3-13 1968.

[2] E. H. Mamdani and S. Assilian, "A fuzzy logic controller for a dynamic plant," Int. J. Man-Machine Study, Vol. 7, No. 1, pp. 1-13, 1975.

[3] M. Sugeno, Industrial applications of fuzzy control, North-Holland, 1985.

[4] G. V.S. Raju, J. Zhou, and R. A Kisner, "Fuzzy logic control for steam generator feedwater control," in Proc. American Control Conference, pp. 1491-1493, San Diego, CA., May 1990.

[5] G. V.S. Raju, J. Zhou, and R. A. Kisner, "Hierarchical fuzzy control," Int. J. of Control, Vol. 54, No. 5, pp. 1201-1216, 1991.

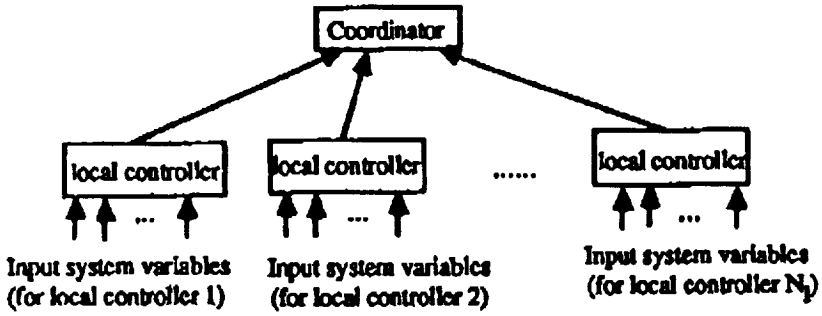


Fig. 1 Hierarchical structure of a controller with a coordinator and several local controllers

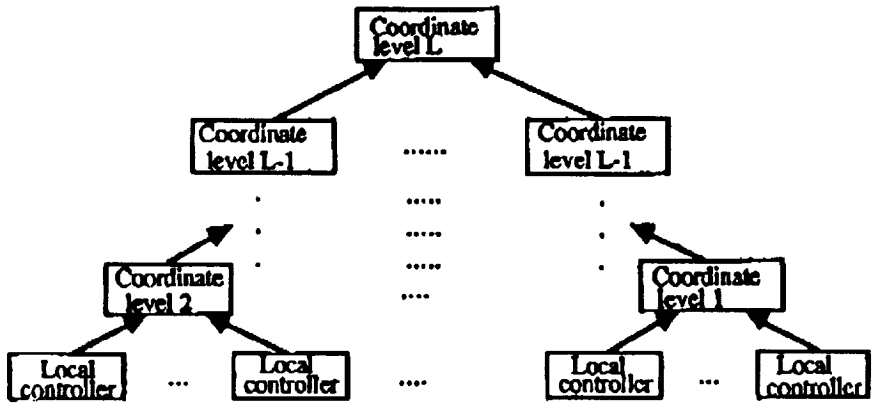


Fig. 2 A general structure of a hierchical controller with several coordinate levels and a local controller level.

Robust Algebraic Image Enhancement for Intelligent Control Systems

Dr. Bao-Ting Lerner
President
KT-Tech, Inc.
7375 Executive Place
Suite 235
Sea brook, M D 20706

and
Michael Morrelli
Optical Systems Laboratory
Department of Electrical Engineering
Texas Tech University
Lubbock, TX 79409

Robust vision capability for intelligent control systems has been an elusive goal in image processing. The computationally intensive techniques a necessary for conventional image processing make real-time applications, such as object tracking and collision avoidance difficult.

In order to endow an intelligent control system with the needed vision robustness, an adequate image enhancement subsystem capable of compensating for the wide variety of real-world degradations, must exist between the image capturing and the object recognition subsystems. This enhancement stage must be adaptive and trust operate with consistency in the present of both statistical and shape-based noise.

To deal with this problem, we have developed an innovative algebraic approach which provides a sound mathematical framework for image representation and manipulation.

Our image model provides a natural platform from which to pursue dynamic scene analysis, and its incorporation into a vision system would serve as the front-end to an intelligent control system.

We have developed a unique polynomial representation of gray level imagery and applied this representation to develop polynomial operators on complex gray level scenes. This approach is highly advantageous since polynomials can be manipulated very easily, and are readily understood, thus providing a very convenient environment for image processing. Our model presents a highly structured and compact algebraic representation of grey-level images which can be viewed as fuzzy sets.

Utilizing the algebraic structure we have devised an innovative, efficient edge detection algorithm, the Lerner Algebraic Edge Detector. This edge detector provides a continuous, single-pixel-wide edge which is a distinct improvement over classical convolution-based edge detectors for enhancing images for object recognition .

Real-time implementation of these algebraic operations on massively parallel architecture can be easily realized due to the parallel characteristics of the polynomial structure as well as the efficient min-max nature of our algebraic system.

Because our algebraic algorithms are highly amenable to high-speed parallel architectures, they have been selected for implementation on a state-of-the-art systolic array processor, the electronically reconfigurable SPLASH Board developed by the Institute of Defense Analysis Supercomputing Research Center (IDA/SRC). In particular, the Lerner Algebraic Edge Detector and the Hough Transform are being ported onto the SPLASH Board to approach a realtime linear feature extraction system .

Based upon our new edge detection scheme, we have developed an accurate method for deriving gradient component information. Moreover, a robust method of linear feature extraction has been derived by combining the techniques of a Hough transform and a line follower. The major advantage of this feature extractor is its general, object-independent nature. Target attributes, such as line segment lengths, intersections, angles of intersection, and endpoints are derived by the feature extraction algorithm and employed during model matching.

N93-22362

Design Issues for a Reinforcement-based Self-Learning Fuzzy Controller

John Yen, Haojin Wang and Walter Dauherity

Center for Fuzzy Logic and Intelligent Systems Research
Department of Computer Science
Texas A&M University
College Station, TX 77843-3112

Fuzzy logic controllers have some often cited advantages over conventional techniques such as PID control: easy implementation, its accommodation to natural language, the ability to cover wider range of operating conditions and others. One major obstacle that hinders its broader application is the lack of systematic way to develop and modify its rules and as result the creation and modification of fuzzy rules often depends on try-error or pure experimentation. One of the proposed approaches to address this issue is self-learning fuzzy logic controllers (SFLC) that use reinforcement learning techniques to learn the desirability of states and to adjust the consequent part of fuzzy control rules accordingly. Due to the different dynamics of the controlled processes, the performance of self-learning fuzzy controller is highly contingent on the design. The design issue has not received sufficient attention. The issues related to the design of a SFLC for the application to chemical process are discussed and its performance is compared with that of PID and self-tuning fuzzy logic controller.

FUZZY EFFICIENCY OPTIMIZATION OF AC INDUCTION MOTORS

Yashvant Jani
Togai InfraLogic, Inc.
Houston, Tx

Gilberto Sousa
University of Tennessee
Knoxville, Tn

Wayne Turner
Research Triangle Institute
Research Triangle Park, N.C.

Ron Spiegel and Jeff Chappell
U.S. Environmental Protection Agency
Research Triangle Park, N.C.

This paper describes the early states of work to implement a fuzzy logic controller to optimize the efficiency of AC induction motor/adjustable speed drive (ASD) systems running at less than optimal speed and torque conditions. In this paper, the process by which the membership functions of the controller were tuned is discussed and a controller which operates on frequency as well as voltage is proposed. The membership functions for this dual-variable controller are sketched. Additional topics include an approach for fuzzy logic to motor current control which can be used with vector-controlled drives. Incorporation of a fuzzy controller as an application-specific integrated circuit (ASIC) microchip is planned.

FUZZY LOGIC CONTROL OF AC INDUCTION MOTORS

In research funded by the U.S. Environmental Protection Agency (EPA), the authors have been pursuing the development of energy optimizer algorithms for ac induction motors driven by adjustable speed drives (ASDs). Our goals are:

1) increase the efficiency of ASD/motor combinations, especially when operating off of rated torque/speed conditions. ASDs using V/Hz control, which is the current predominant industry standard, still do not gain maximum efficiency from motors operating at less than rated loads and speeds;

2) develop a generic energy efficiency optimizing controller (EEOC) which can be applied to a wide range of ac induction motors, regardless of their size and corresponding equivalent circuit values;

3) develop an energy efficiency optimizing controller (EEOC) which can eliminate the requirement for tachometer or encoder feedback, and still maintain the stability of closed-loop control; and

4) develop an energy efficiency optimizing controller (EEOC) which is self-tuning, thus eliminating the need for extensive operator/manufacturer involvement in the installation of the energy optimizer into ASDs.

Fuzzy logic approaches to these goals are attractive for two reasons:

1) the use of fuzzy logic promises to simplify the energy efficiency optimizing controler (EEOC) control problem, which is highly nonlinear;

2) fuzzy logic offers a way to develop an energy efficiency optimizing controller (EEOC) controller which will offer the stability of closed loop control without the need for speed feedback, thus eliminating the cost of tachometers and encoders.

Fuzzy Efficiency Optimization for Steady State Motor Operation

Our main interest has been to solve the problems above for large horsepower motors (>10 hp) running at steady state conditions in industrial applications (e.g. pump and fan motors).

An induction motor simulator has been developed based on the equivalent circuit representation of a motor. As a starting point, the simulation values which are produced correspond to those which would be produced by a V/Hz controller. The simulator computes the values of the motor state variables (currents, voltages, power, frequency, etc.) in response to changes in the value of the stator voltage, V_s. The values of V_s are

provided to the simulator by a fuzzy energy optimizer. (This energy optimizer was discussed in a previous paper delivered at FUZZ-IEEE '92 in San Diego in March.) This energy optimizer, referred to as the Single Variable Fuzzy Logic Motor Controller, and illustrated in the accompanying block diagram, alters the value of stator voltage (V_s) and then measures the input power P_{in} to see if it has changed.

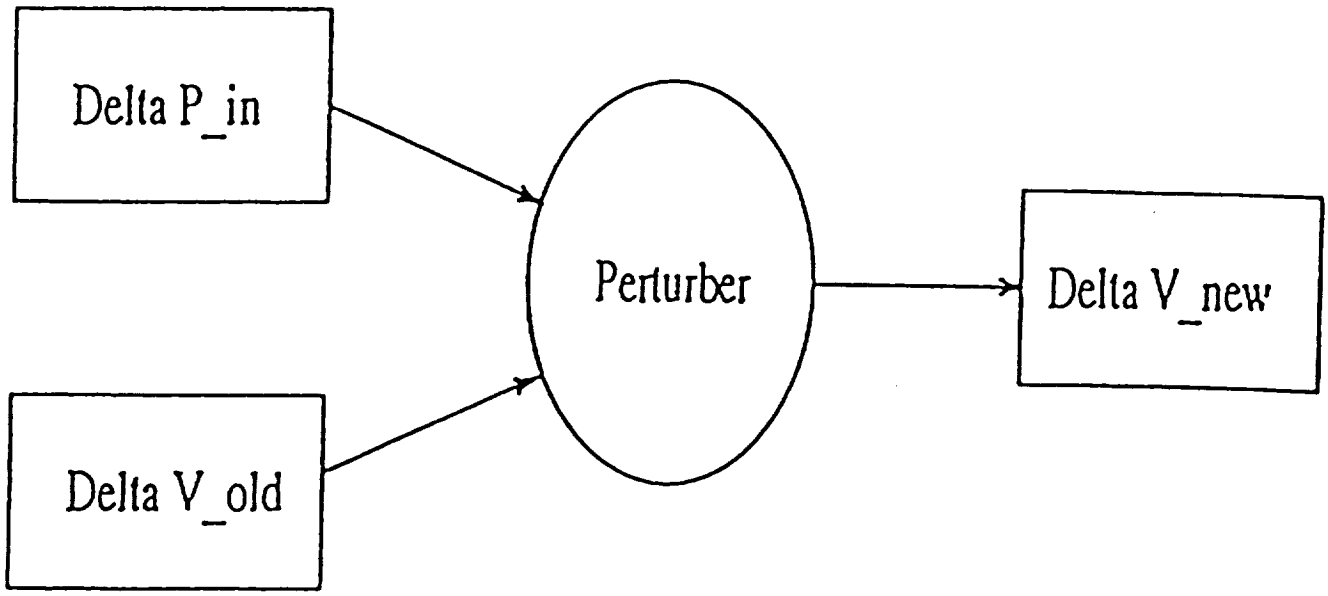


Figure 1. Block Diagram of the Fuzzy Logic Energy Optimizer

Dependent on the magnitude and direction of the change in P_{in}, a set of fuzzy rules, represented here by the section labeled 'Perturber' in the block diagram and using ΔP_{in} and the last change in V_s, ΔV_{s_old}, as inputs, computes an incremental change in the stator voltage ΔV_{s_new} which is then applied to the simulator. A new set of state variables is computed and the process is repeated until either a minimum input power is obtained, characterized by the return of a value of 0 for ΔV_s from the fuzzy controller, or, if tolerance limits on the output torque or the shaft speed of the motor have been exceeded. After some testing, the max-dot inference method and centroid defuzzification were employed.

This technique is essentially a search scheme for the minimum input power point, which occurs in a motor driven by a pulse-width-modulation (PWM) ASD when the copper losses and core losses of the motor are equivalent, as shown in the following figure.

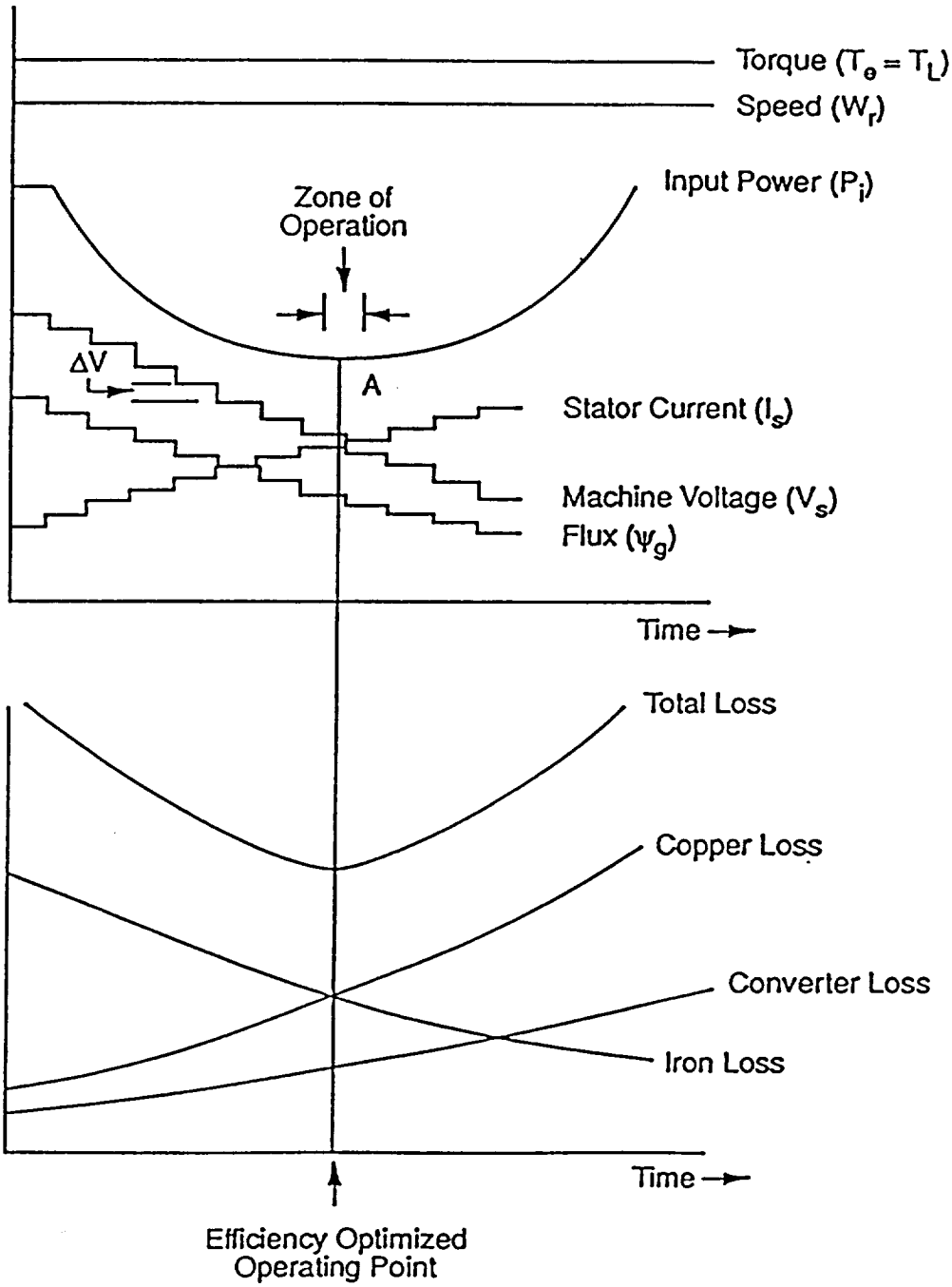


Figure 2. Efficiency Optimization Control based on Real-time Search.

Note that the prediction in the search scheme is that the stator voltage will decrease and the stator current will increase. This prediction has been borne out by the simulator results. The simulator also predicts efficiency improvements by the energy efficiency optimizing controller (EEOC) over standard V/Hz control, as shown in Figure 3.

After the controller rules were refined from simulation of motors of various sizes, a set of 13 fuzzy rules were developed, shown in Table 1.

RULES

1. IF ΔP_{in} IS N AND ΔV_{s_old} IS N, THEN ΔV_{s_new} = N.
2. IF ΔP_{in} IS N AND ΔV_{s_old} IS P, THEN ΔV_{s_new} = P.
3. IF ΔP_{in} IS N AND ΔV_{s_old} IS NM, THEN ΔV_{s_new} = NM.
4. IF ΔP_{in} IS N AND ΔV_{s_old} IS PM, THEN ΔV_{s_new} = PM.
5. IF ΔP_{in} IS NM AND (ΔV_{s_old} IS NM OR ΔV_{s_old} IS N), THEN ΔV_{s_new} = NM.
6. IF ΔP_{in} IS NM AND (ΔV_{s_old} IS PM OR ΔV_{s_old} IS P), THEN ΔV_{s_new} = PM.
7. IF ΔP_{in} IS PM AND (ΔV_{s_old} IS NM OR ΔV_{s_old} IS N), THEN ΔV_{s_new} = PM.
8. IF ΔP_{in} IS PM AND (ΔV_{s_old} IS PM OR ΔV_{s_old} IS P), THEN ΔV_{s_new} = NM.
9. IF ΔP_{in} IS P AND ΔV_{s_old} IS NM, THEN ΔV_{s_new} = PM.
10. IF ΔP_{in} IS P AND ΔV_{s_old} IS PM, THEN ΔV_{s_new} = NM.
11. IF ΔP_{in} IS P AND ΔV_{s_old} IS N, THEN ΔV_{s_new} = P.
12. IF ΔP_{in} IS P AND ΔV_{s_old} IS P, THEN ΔV_{s_new} = N.
13. IF ΔP_{in} IS Z AND ΔV_{s_old} IS ANY, THEN ΔV_{s_new} = Z.

Table 1. Single Variable Controller Rules

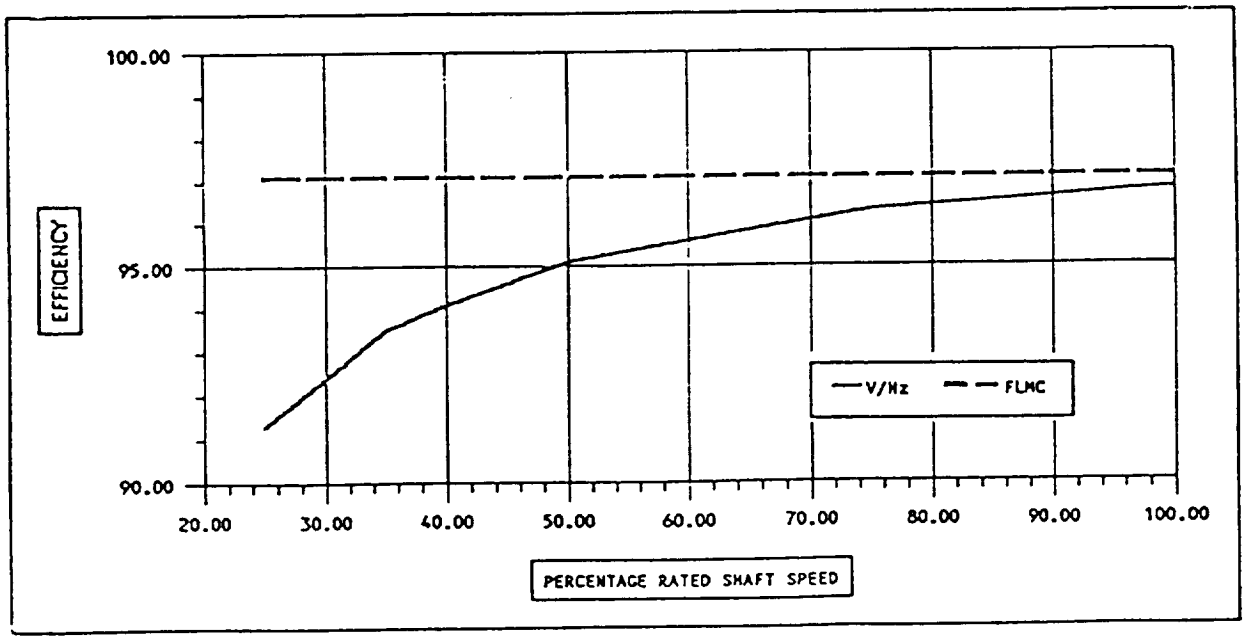


Figure 3. **Energy Efficiency Optimizing Control (EEOC) versus V/Hz Control for a 100 HP Motor with Torque $\propto$ Speed^2.**

The variable values, N, NM, P, PM, and Z stand respectively for negative, negative medium, positive, positive medium, and zero. Data gathered from the motor simulator led to development of limits for membership functions for the fuzzy variables voltage and power. Figure 4 illustrates this for the linguistic variable ΔP_{In}.

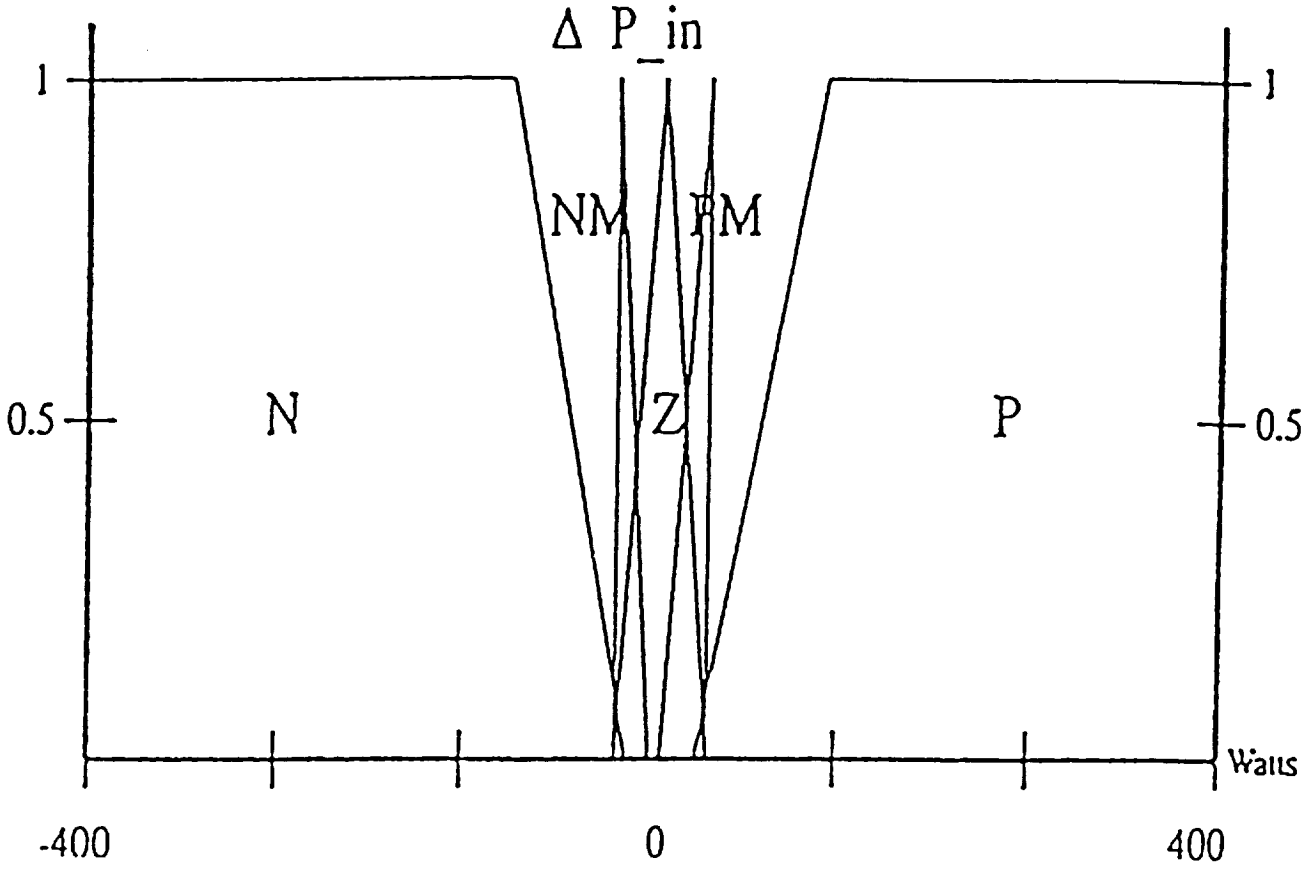


Figure 4. Final Membership Functions for the Fuzzy Variable ΔP_{in}.

It was found from the simulator that the P_{in} can vary by as much as ±400W. Surfaces were constructed from curves relating the various changes in ΔV_{new} to changes in ΔP_{in} and ΔV_{old}. An example of such a surface, generated from data collected with the simulator, is shown in Figure 5.

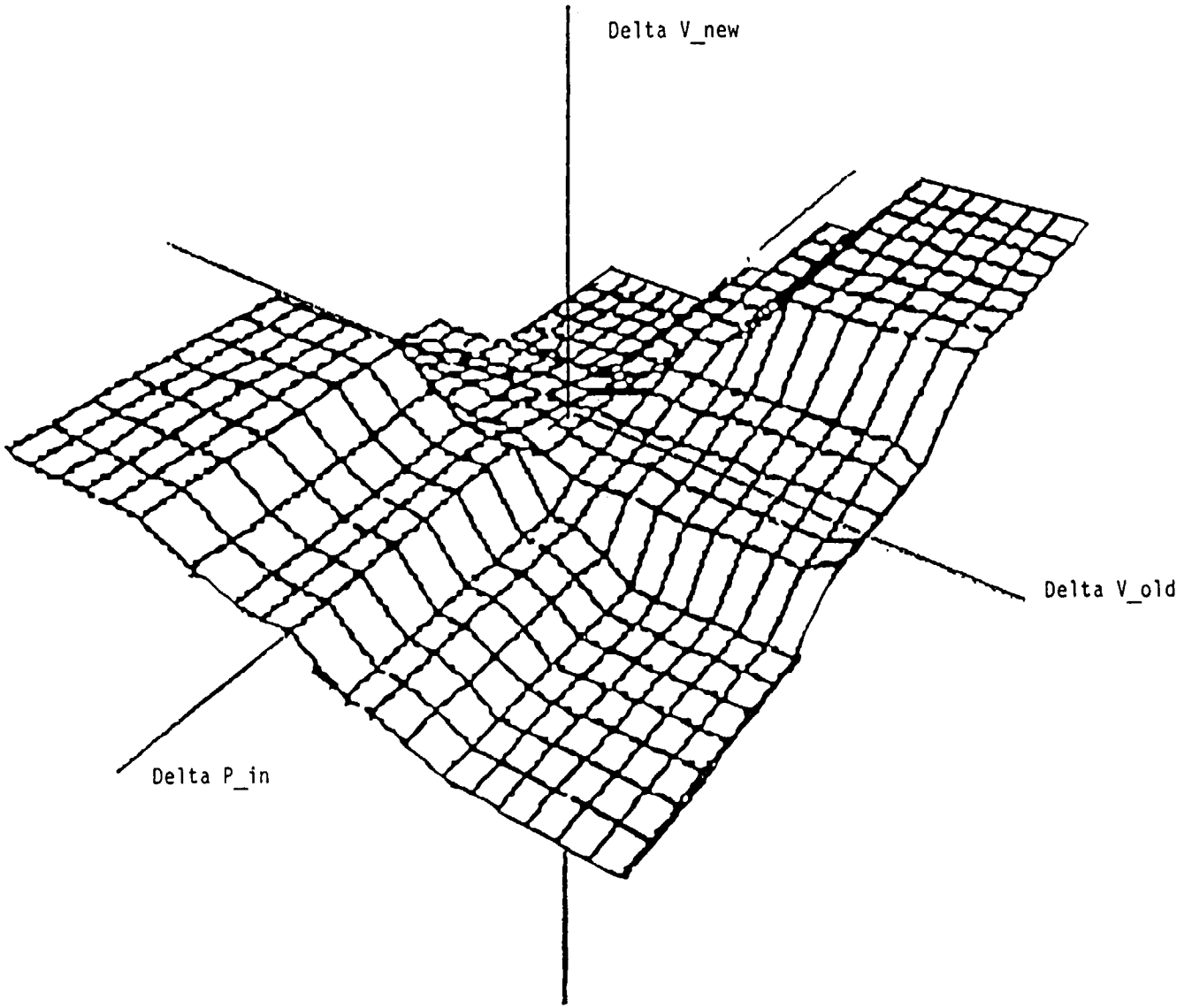


Figure 4. Surface Generated from Simulator Data Relating Changes in ΔV_{new} to Changes in ΔP_{in}.

These surfaces can be used to optimize the membership functions by examining the surfaces for abrupt or discontinuous changes in the output variable ΔV_s at various values of the input variables ΔP_{in} and ΔV_{s_old}. Based on the magnitude of the discontinuity either the input membership functions' overlaps could be changed or the width of the output membership functions could be changed.

As this initial controller was used to simulate, from equivalent circuit data, several different motors, several features of the controller became clear as this data was analyzed. For example, any change in stator voltage produced a drop in the output shaft speed ϖ_r, which is generally undesirable. Also, for a given set of equivalent circuit values, maximum efficiency is closely related to total circuit impedance, Z_{in} , regardless of the torque/speed condition.

Because of the loss of shaft speed, it was clear that even the optimized controller would never perform adequately working alone. Therefore attention was turned to a controller which could both compensate for the loss in shaft speed resulting from the voltage perturbations and still allow a minimum input power point to be reached. It was recognized that the loss of rotor speed could be corrected by increasing the frequency of the stator voltages and currents, while the minimum power input point can be obtained by perturbing the voltage. Furthermore, a correlation of ω_r impedance suggested that if the change in input impedance were known for a particular change in synchronous frequency ω_e and voltage V_s, then approaching an optimum impedance as rapidly as possible should achieve both the minimum input power and the correction of the drop in ϖ_r. This led us to develop a preliminary controller concept for a frequency perturber, shown here in block diagram form in conjunction with the existing voltage perturber.

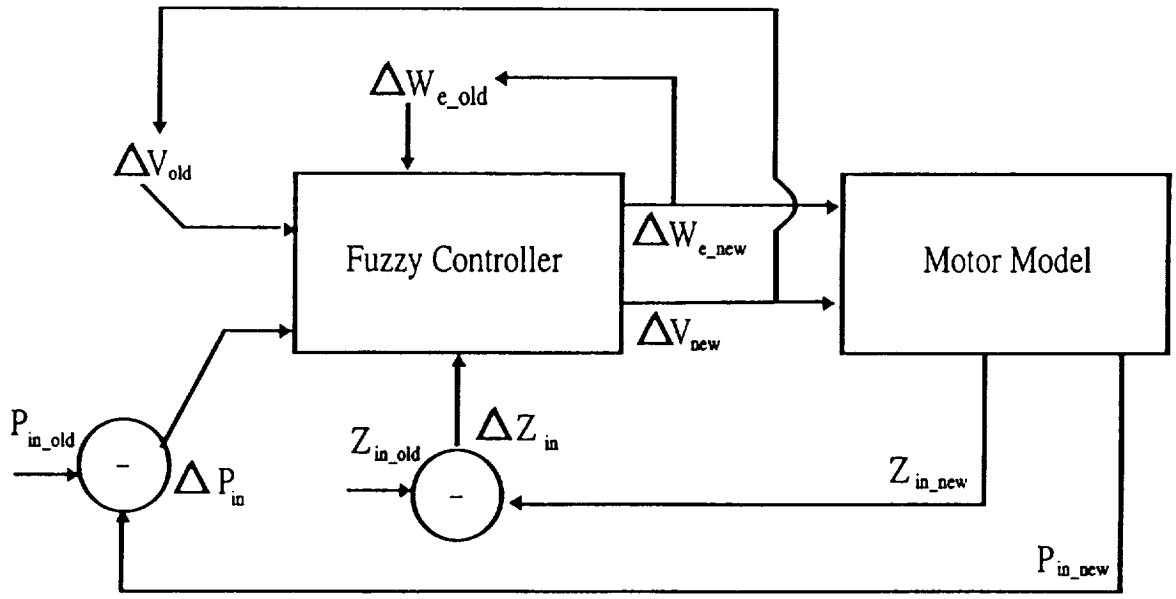


Figure 6. Dual Variable Fuzzy Logic Controller for AC Induction Motor

Thus the set of rules which perturbed the voltage were augmented by another set of rules which perturbed ω_e using the previous value of $\Delta\omega_e$, $\Delta\omega_{e\text{-}old}$, and ΔZ_{in}. This new fuzzy rulebase, which has 9 rules, fires simultaneously with the 13 rules of the SVFLC. The rule-base for inference of the synchronous frequency is shown in the following table.

RULES	
1)	IF $\Delta\omega_{e\_old}$ IS P AND ΔZ_{in} IS N THEN $\Delta\omega_e$ = P
2)	IF $\Delta\omega_{e\_old}$ IS Z AND ΔZ_{in} IS N THEN $\Delta\omega_e$ = P
3)	IF $\Delta\omega_{e\_old}$ IS N AND ΔZ_{in} IS N THEN $\Delta\omega_e$ = N
4)	IF $\Delta\omega_{e\_old}$ IS P AND ΔZ_{in} IS Z THEN $\Delta\omega_e$ = Z
5)	IF $\Delta\omega_{e\_old}$ IS Z AND ΔZ_{in} IS Z THEN $\Delta\omega_e$ = Z
6)	IF $\Delta\omega_{e\_old}$ IS N AND ΔZ_{in} IS Z THEN $\Delta\omega_e$ = Z
7)	IF $\Delta\omega_{e\_old}$ IS P AND ΔZ_{in} IS P THEN $\Delta\omega_e$ = N
8)	IF $\Delta\omega_{e\_old}$ IS Z AND ΔZ_{in} IS P THEN $\Delta\omega_e$ = N
9)	IF $\Delta\omega_{e\_old}$ IS N AND ΔZ_{in} IS P THEN $\Delta\omega_e$ = P

Table 2. Added Rules for Control Frequency

The symbols P, N, and Z stand respectively for positive, negative and zero. Limits on the membership functions were developed as before by analyzing output data from the simulator and setting the limits. The preliminary output membership functions for $\Delta\omega_e$ are illustrated in the following figure.

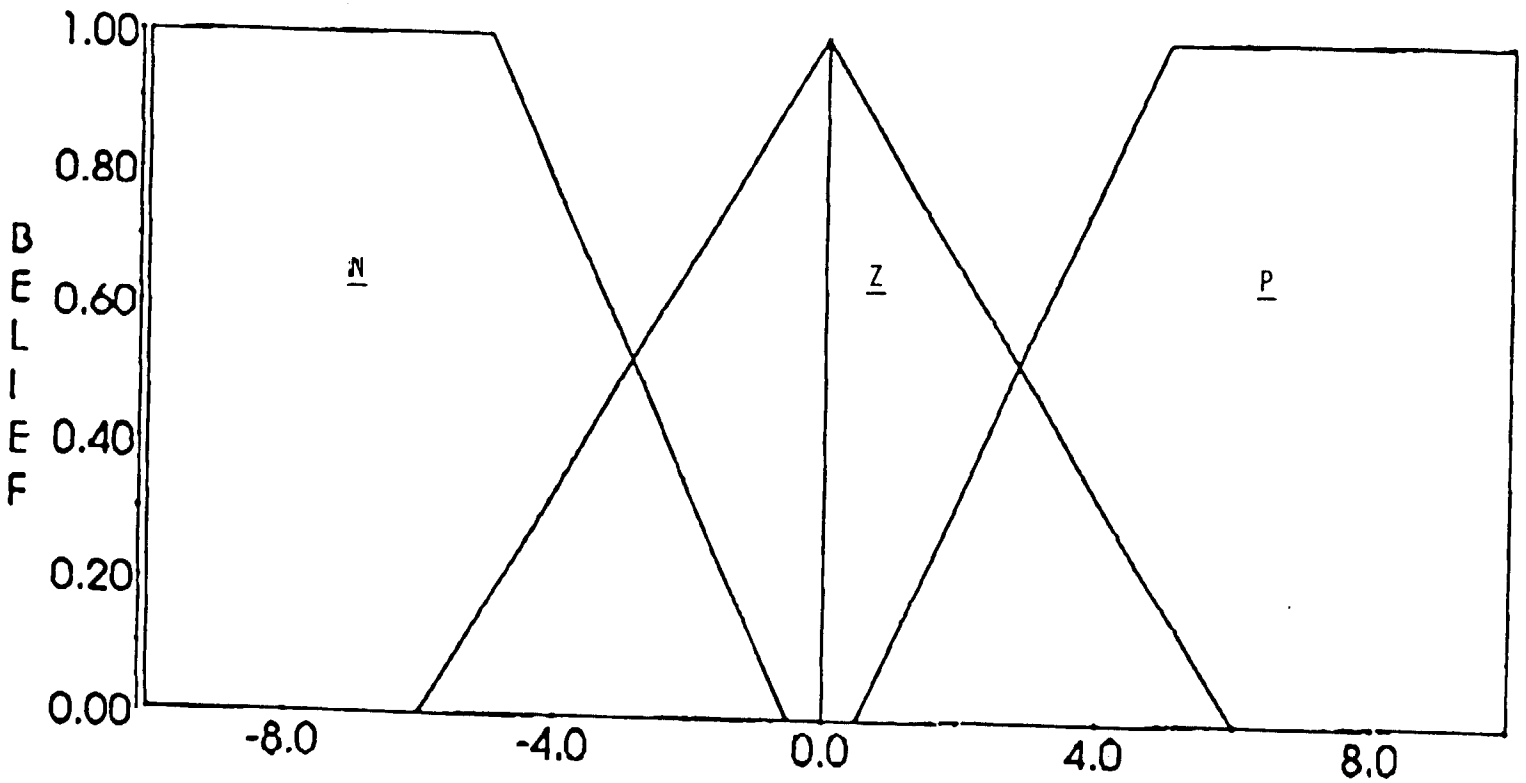


Figure 7. Membership Functions for ΔW_e

As in the previous work, a control surface was generated and could be used to tune the membership functions. Note in the control surface graph that there are certainly abrupt changes in $\Delta\omega_e$ for certain values of ΔZ_{in} and $\Delta\omega_{e\ old}$.

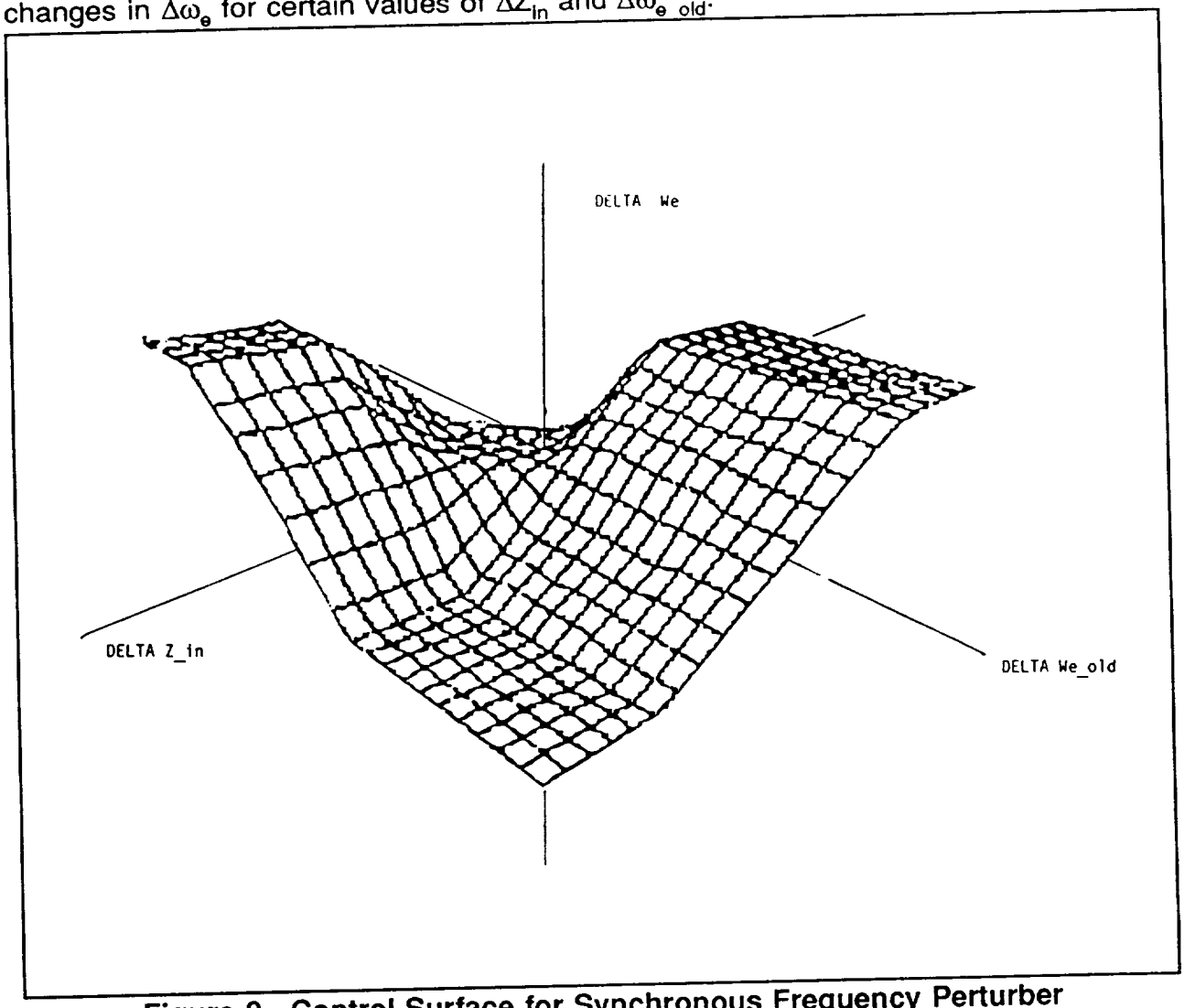


Figure 8. Control Surface for Synchronous Frequency Perturber

Fuzzy Efficiency Optimization using Indirect Vector Control

A parallel effort is taking place to provide fuzzy efficiency optimization for induction motors which use indirect vector or field-oriented control of induction motors rather than PWM. Indirect vector control is another approach to the control of ASD/motor combinations which controls current rather than voltage. This type of energy optimizer emphasizes the suppression of transient phenomena in the motor, and is focused more on dynamic process control applications (lathe motors, steel mill rolling, etc) than the steady state controller. The controller is illustrated in Figure 9.

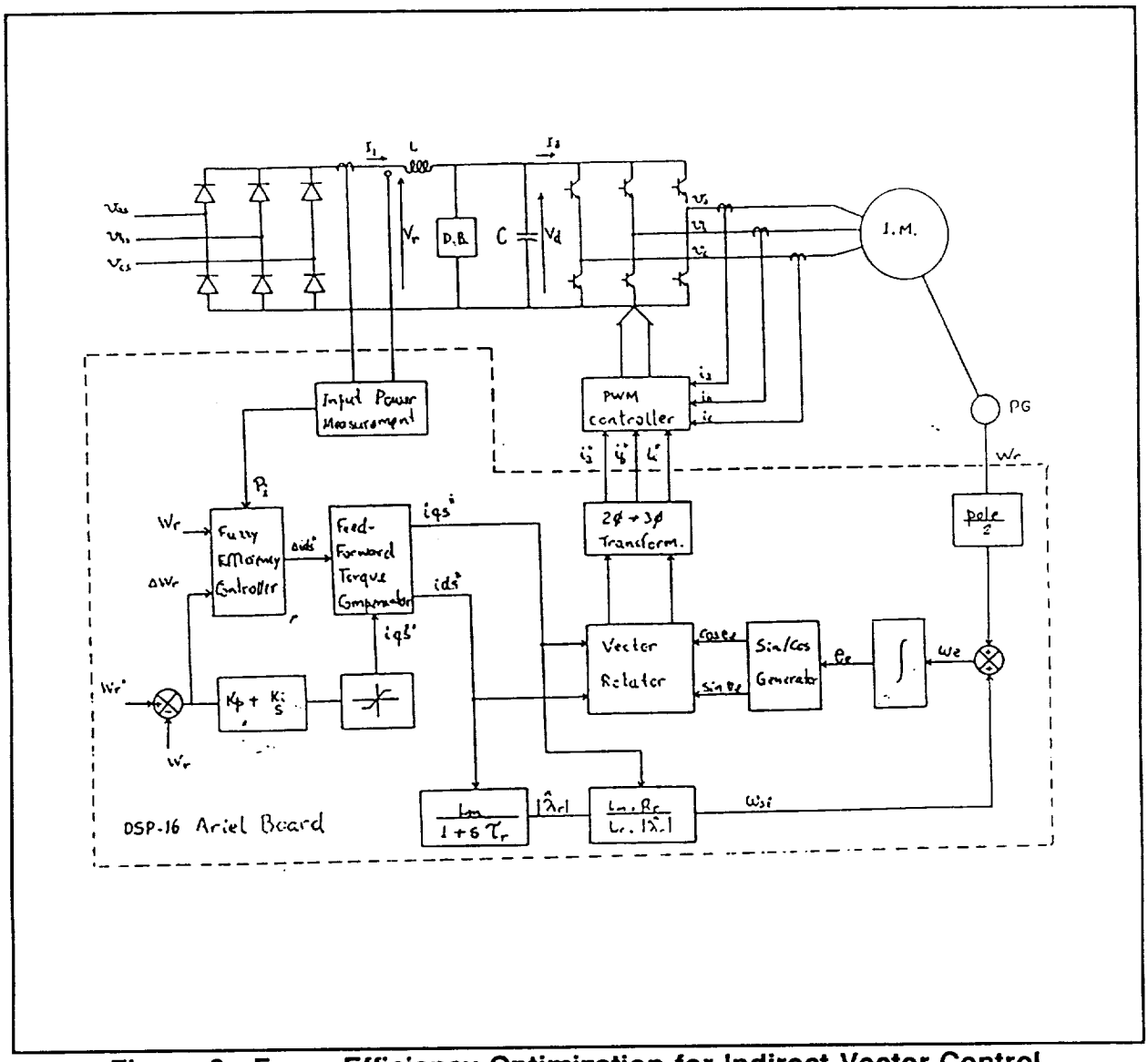


Figure 8. Fuzzy Efficiency Optimization for Indirect Vector Control.

In indirect vector control, the motor is modeled using a change of variables which represents the state variables of the motor in terms of two magnetically decoupled equivalent circuits, generally referred to as the d-q representation of a motor. When vector control is employed the currents i_{ds} and i_{qs} control the flux and the torque of the machine, respectively.

Fuzzy efficiency optimization for indirect vector control utilizes the same type of minimum input power search scheme outline above, however rather than perturbing the stator

voltage, the rotor flux λ_r is changed by perturbing the current i_{ds}. Then P_{in} is measured to see if the input power has changed. In the event that it has, a set of fuzzy rules computes a new value of Δi_{ds}, based on ΔP_{in} and the previous value of Δi_{ds}, referred to as $L\Delta i_{ds}$. Then P_{in} is measured again and the process is repeated. A table showing the preliminary rules relating Δi_{ds} to ΔP_{in} and $L\Delta i_{ds}$ is shown in the following table.

RULES

1. If LDids is N and ΔPi is PB, then LDids is PB.
2. If LDids is N and ΔPi is PM, then LDids is PM.
3. If LDids is N and ΔPi is PS, then LDids is PS.
4. If LDids is N and ΔPi is ZE, then LDids is ZE.
5. If LDids is N and ΔPi is NS, then LDids is NS.
6. If LDids is N and ΔPi is NM, then LDids is NM.
7. If LDids is N and ΔPi is NB, then LDids is NB.
8. If LDids is P and ΔPi is PB, then LDids is NB.
9. If LDids is P and ΔPi is PM, then LDids is NM.
10. If LDids is P and ΔPi is PS, then LDids is NS.
11. If LDids is P and ΔPi is ZE, then LDids is ZE.
12. If LDids is P and ΔPi is NS, then LDids is PS.
13. If LDids is P and ΔPi is NM, then LDids is PM.
14. If LDids is P and ΔPi is NB, then LDids is PB.

Table 3. Fuzzy Rules for Efficiency Optimization with Indirect Vector Control.

A total of 14 IF-THEN rules are defined for the energy optimizer utilizing indirect vector control.

Figure 10 illustrates the preliminary membership functions derived from observation of results obtained from computer simulations.

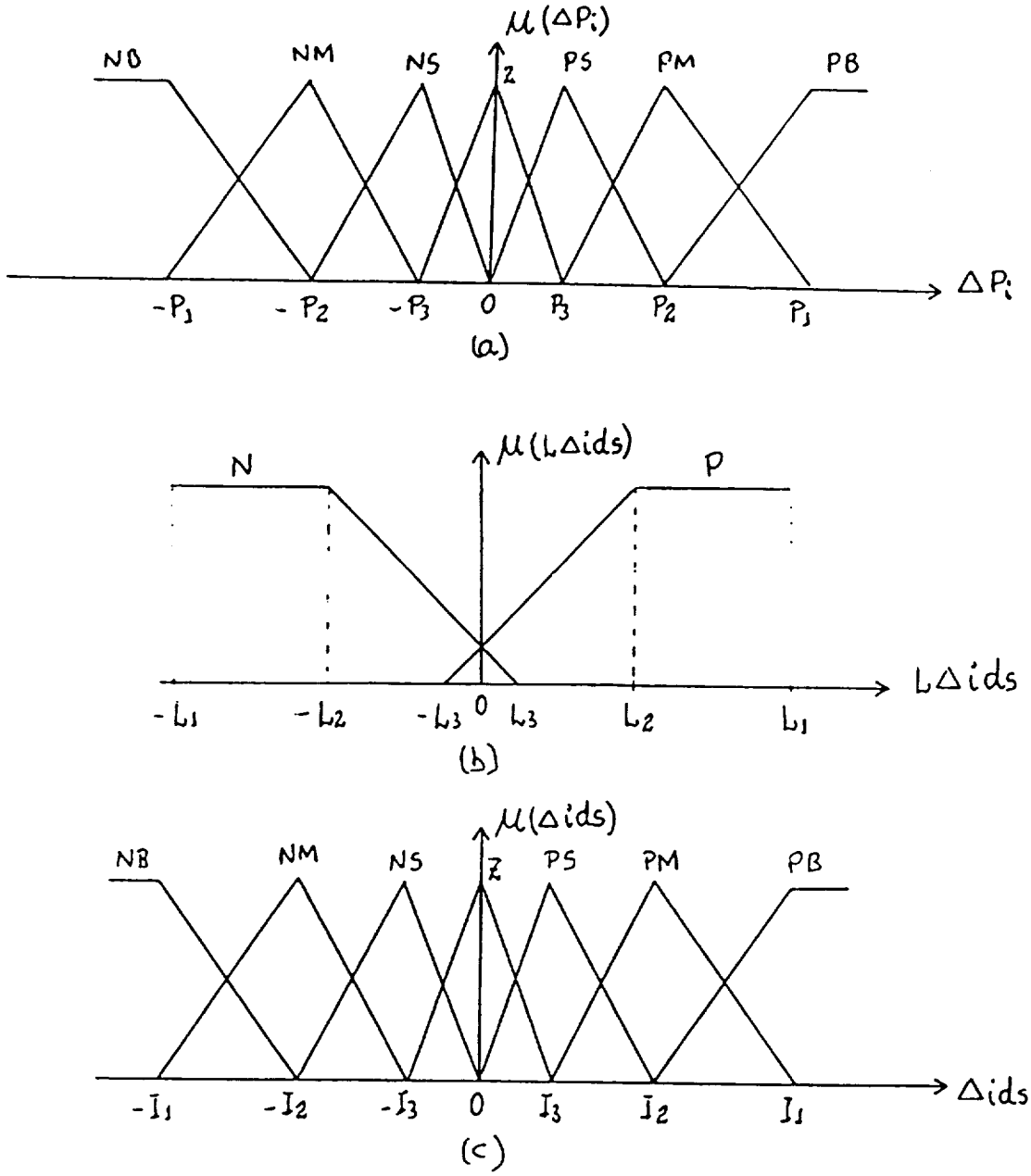


Figure 10. **Preliminary membership functions for fuzzy efficiency controller: (a) change in input power; (b) last change in current i_{ds}; (c) new change in current i_{ds}.**

The membership functions were developed using variables normalized in the interval [-1,1], hence the magnitudes of the endpoint variables (+/- P_1, +/- L_1, +/- I_1)across the domain of the membership functions is 1. The values of the interior limits on the membership functions have not at present been arrived at.

The max-min method of inference is being applied to obtain truth values of any particular rule, hence the design of the fuzzy membership functions for $L\Delta i_{ds}$ provides a degree of limitation for the truth value of a rule when $L\Delta i_{ds}$ is "negative small" or "positive small", even though there is no membership function specifically for those fuzzy values. This avoids using multiple membership functions in a place where fewer will perform the same job, and thereby reduces the size of the fuzzy rulebase. The overlap between the positive and negative membership functions assure that division by 0 will not occur in the height defuzzification method used by UT, since even if $L\Delta i_{ds}$ is 0, it will have a non-zero degree of belief in either the 'P' or 'N' region.

Reducing the flux to achieve minimum input power has an effect similar to that of reducing voltage in the previous controller. The shaft speed will drop. We have found that this can be compensated for by a change in the torque component of current i_{qs}. This is a function of the change in i_{ds}. After a change in the value of i_{qs} is made (which is not a fuzzy operation) fuzzy efficiency optimization is not reapplied until the machine has returned to steady-state condition, which is determined by comparing the sum of the absolute values of the last three rotor speed errors ($\Delta\omega_{r)}$ to a tolerance value of 1 rad/sec. At that time a new value of Δi_{ds} is computed by the fuzzy efficiency optimizer and the cycle repeats. Even after optimum efficiency has been reached, this steady-state condition is checked for periodically in order to determine that no process disturbance has taken place which would require the controller to act in order to produce the required torque output or required speed.

All rules and membership functions are being tuned using computer simulation and by testing the controller in a laboratory setting. The following diagram shows the overall scheme of the laboratory setup.

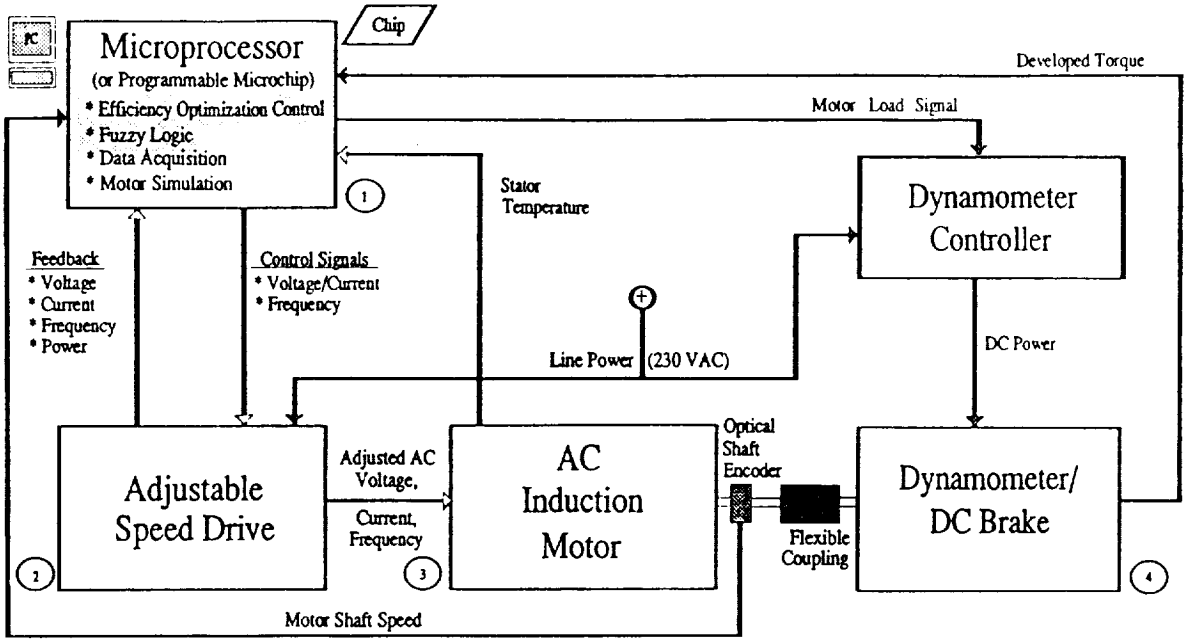


Figure 11. Motor Laboratory System

The fuzzy rules are executed in a 486/33 MHz from code compiled with other C routines, which also monitor system information via a data acquisition board, and communicate with the ASD to alter the ASD voltage and frequency output. The same code also directs an analog output on the data acquisition board to vary the strength of the field in the DC brake via the dynamometer controller, thus simulating various degrees of load on the motor.

Summary

Computer simulations have shown that a fuzzy controller which optimizes the use of energy by a motor/ASD combination can be developed. To be truly effective, the controller should alter both the stator voltage and stator frequency while maintaining the output power required of the motor/ASD system for the drive at hand. Energy efficiency optimization can be applied not only to drives which produce sinusoidal PWM output, but to indirect-vector controlled drives as well.

N93-22364

ADVANCED TELEROBOTIC CONTROL USING NEURAL NETWORKS*

Robert M. Pap[1,2]; Mark Atkins [1,2]; Chadwick Cox [1,2];
Charles Glover [2,4]; Ralph Kissel [3] and Richard Saeks [1,2,5]
1 Accurate Automation Corp; P.O.Box 11295; Chattanooga, TN 37401
2 Tennessee Center for Neural Engineering @ Tennessee State Univ.
3 NASA, George C. Marshall Space Flight Center
4 Oak Ridge National Laboratory
5 Illinois Institue of Technology, Armour School of Engineering

Accurate Automation is designing and developing an adaptive decentralized joint controllers using neural networks. We are then implementing these in hardware for the Marshall Space Flight Center PFMA as well as to be usable for the Remote Manipulator System (RMS) robot arm. Our design is being realized in hardware after completion of the software simulation. This is implemented using a Functional-Link neural network.

We have completed the theoretical analysis and design of the neural network based Decentralized Adaptive Joint Controller (DAJC) for use in a telerobotic system including, evaluating appropriate neural network architectures and learning rules, as well as a mathematical and empirical robustness analysis.

We are building an neural net based controller to model the arm dynamics and generate the necessary joint torques. This neural network receives position and velocity information from the joint encoders and provides a mapping from this space to the joint-torque space of the arm. Our neuro-joint controller produces a trajectory in joint-torque space. This trajectory represents the sequence of torques values as a function of time that are necessary to perform a given task.

Our tests to date have on a simulated Puma 560 robot arm and a NASA RMS robot arm run on a Silicon Graphics 4D340VGX SuperWorkStation. The simulation modeled inertia, interjoint couplings, centrifugal torque, gravity loading, and viscous and Coulomb frictions. This is a three joint simulation. This is now being interfaced to the Extendable Stiff Arm Manipulator (ESAM) robot arm from NASA Marshall to validate the simulation before the neuro joint controller is translated to a hardware implementation. In the future we will test our neurocontroller on the PFMA or RMS.

In the simulation testing that we have done so far, we have controlled three joints and moved the arm from various starting positions to various ending positions in different amounts of time. We have also subjected the controller to variable amounts of mass on the end-effector, including masses dropped on the arm during the move and taken off the arm before the move was complete. Oscillations tend to damp out quickly as the controller compensates for error. The controller has proven to be very stable and robust. End effector mass changes of 90kg can be handled during a two second move from (-180,0) to (0,90) in 1 g.

This work is supported by the National Aeronautics and Space Administration through an Small Business Innovation Research Phase II contract administered by the George C. Marshall Space Flight Center under contract NAS8-38967. The authors wish to express their appreciation to Dr. Clifford Parten of the University of Tennessee at Chattanooga, Dr. Paul Werbos of the National Science Foundation, Dr. Joel Davis of the Office of Naval Research and Dr. Thomas Bryant and Harry Johnson of NASA. Additional support for this project has been provided by the Office of Naval Research and the National Science Foundation.

Multi-layered Reasoning by means of Conceptual Fuzzy Sets

Tomohiro Takagi, Atsushi Imura, Hirohide Ushida and Toru Yamaguchi*

Laboratory for International Fuzzy Engineering Research (LIFE)
Siber Hegner Building 3FL.,89-1 Yamashita-cho, Naka-ku, Yokohama-shi 231 JAPAN

Key Words: context dependency, fuzzy sets theory, fuzzy associative memory, approximate reasoning, neural network, knowledge representation, inductive learning, concept formation

1. Introduction

Real world consists of a very large number of instances of events and continuous numeric values. On the other hand, people represent and process their knowledge in terms of abstracted concepts derived from generalization of these instances and numeric values. Logic based paradigms for knowledge representation use symbolic processing both for concept representation and inference. Their underlying assumption is that a concept can be defined precisely. However, as this assumption hardly holds for natural concepts, it follows that symbolic processing cannot deal with such concepts. Thus symbolic processing has essential problems from a practical point of view of applications in the real world. In contrast,fuzzy set theory can viewed as a stronger and more practical notation than formal, logic based theories because it supports both symbolic processing and numeric processing, connecting the logic based world and the real world.

For example, in the case of an intelligent control system, control actions are determined not only by numeric processing but also integrated with the result of intellectual decision making at a more abstract level based on meaning understanding of numeric data. Using only numeric processing or describing simple correspondences of instances produces a black box effect and is difficult to integrate with symbolic, logic based information processing. For this reason, multi-layer structured frameworks have been proposed, where intellectual information processing based on meaning understanding and state recognition in upper layer supervises the data processing in lower layer [2]-[3]. The duality abstract/concrete of the real world is reflected in the intelligent/lack of intelligence duality at the intellectual level (Increasing Precision with Decreasing Intelligence principle, IPDI, [4] - [5]) To cope with this duality a knowledge representation paradigm must be able to hierarchically represent both aspects. Thus we are led to consider multi-layered structures representation.

A concept such as an operator's know-how in the upper abstracted layer is essentially vague. Moreover, it is difficult to eliminate this vagueness during the generalization process from control experiences. For this reason, fuzzy set theory can be expected to provide us with a strong notation for concept representation at different levels of granularity: lower, concrete concepts describe an upper, vague concept constructing thus a multi-layered structure and a capability connecting information processing in different layers.of abstraction.

However, simple notion using ordinary fuzzy sets cannot solve all the problems of (concept) knowledge representation because of the following:

---

* Currently at Systems & Software Engineering Laboratory, Toshiba Corp.

1. Lack of context dependency
2. Impossibility of explicit formulation of a concept.

These problems arise because the meaning of a concept changes depending on various situations and concrete events cannot always be generalized into logical notation explicitly. For example, a fuzzy controller of a car aims to realize intelligent control in terms of modeling the driver's know-how such as : "If the distance between cars is big, then the change of acceleration is big". Nevertheless, since the concepts such as "big" or "small" describing control rules are defined on a numeric axis absolutely using a simple formulation, the definition indicates only a simply unique meaning of a concept and cannot cover the variety of meanings (depending on the size of a car and road conditions). The fuzzy control does not achieve the intellectual information processing in the upper level nor the aims of intelligent control.

All these problems relate to the representation of **the meaning** of a concept. According to Wittgenstein [1], the meaning of a concept is represented by the totality of its uses. In this spirit we proposed [2] the notion of Conceptual Fuzzy Sets:(henceforth referred to as CFS). In the CFS the meaning of a concept is represented by the distribution of activation of labels naming concepts. Since the distribution changes depending on the activated labels to indicate a situation, CFS can represent context dependent meanings. CFS are realized using bidirectional associative memories implemented as neural networks. Since the propagation of activation realizes logical operations and inference as well as the representation of meanings, many advantageous features are obtained which are not realized by logic based representation alone.

Further, since the distribution of activation determined by the propagation of activation in CFS represents the meaning of a concept, the propagation of activations corresponds to reasoning. In particular, a multi-layer structured CFS represents the meaning of a concept in various expressions in each layer. Therefore, it follows that due to the capability of naturally realizing information processing in multi-layered structures, the CFS have the following features:

1. Because CFSs are realized and connected using a bi-directional associative memory, CFS can carry out information processing both in the upper layer and lower layer simultaneously exchanging information. Thus they provide us easily with a framework where the processing in the upper layer supervises the processing in the lower layer.
2. Since CFS are realized as a bi-directional associative memory, it can carry out both bottom-up processing from the lower layer to the upper layer, and top-down processing from the upper layer to the lower layer simultaneously.

In this paper, we propose Multi-layered Reasoning realized by using CFS and we discuss the above two features. In section 2, we show the general characteristics of CFS. In section 3, we discuss the structure where the upper layer supervises the lower layer and we illustrate it with examples. In section 4, we discuss the context dependent processing carried out by the simultaneous bottom-up processing and top-down processing.

2. Conceptual Fuzzy Sets

2.1. Conceptual Fuzzy Sets for Concept Representation

A label of a fuzzy set represents the name of a concept and a fuzzy set represents the meaning of the concept. Therefore, the shape of a fuzzy set should be determined from the meaning of the label depending on various situations. According to the theory of meaning representation from use proposed by Wittgenstein [7], the various meanings of a label (word) may be represented by other

labels (words) and we can assign grades of activation showing compatibility degrees between different labels.

The Conceptual Fuzzy Set proposed in [8], achieves this by the distributions of activations. Since the distribution changes depending on the activated labels which indicate conditions, the activations resulted through CFS show a context dependent meaning. When more than two labels are activated CFS is realized by the overlapping propagations of activations. In CFS notations, operations and their controls are all realized by the distributions of activation and their propagations in associative memories.

We can say that the distribution determined by the activation of a label agrees with the region of thought corresponding to the word expressing its meaning. Since situations are also indicated by activations, the meaning is expressed by overlapping the regions of thought determined by these activations. Fig 2.1 illustrates the different meanings of the same label, L1, in different situations, S1 and S2.

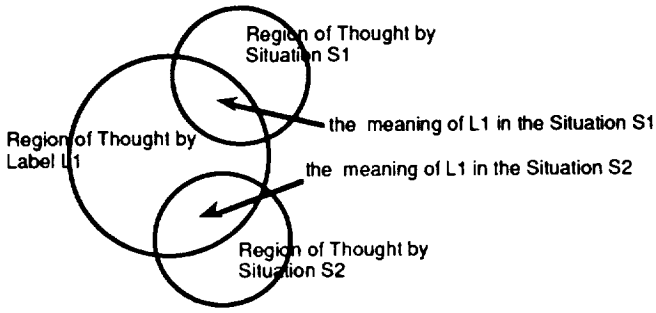


Fig.2.1 Different meanings in different situations

A CFS is realized as an associative memory, in which a node represents a concept and a link represents a strength of the relation between two (connected) concepts. Activations of nodes produce a reverberation and the system energy is stabilized to a local minimum where corresponding concepts are recollected as a result. The recollections are carried out through a weight matrix encoded from stimulus-response paired data.

In this paper we use Bidirectional Associative Memories (BAMs) [9] because of the clarity of constraints for their utilization. At the association in BAMs reverberations are carried out according to:

$$Y_t = \phi(M \cdot X_t), \quad X_{t+1} = \phi(M^T \cdot Y_t). \qquad (1)$$

where, $Xt=[x1, x2, ..., xm]^T$, $Yt=[y1, y2, ..., yn]^T$ are activation vectors on x and y layers at the reverberation step t, and $\phi(\cdot)$ is a sigmoid function of each neuron. BAMs memorize corresponding pairs of elements at each layer in terms of a synaptic weight matrix, M, to memorize CFS, and calculated from corresponding input/output pairs of Ai /Bi with coefficient α i:

$$M = \Sigma i \ \alpha i Ai'Bi' \qquad (2)$$

Example 2.1. CFS representing a composed concept which has multiple meanings depending on situations

Let us consider the concept "tall", and its meaning according to whether it is applied to an American or Japanese person. The meaning of concept "tall" changes in these two situations. The distribution

of activation of other labels explains the meaning of "tall" depending on these contexts. Fig. 2.2 shows the concept "tall American" which agrees with the meaning of "tall" in case of an American person. In this figure and throughout the remainder of this paper, "American" and "Japanese" refer to "American height" and "Japanese height" respectively. The activations of nodes which express "American" and "tall" make the distribution of activation in the middle layer which consists of numerical values.

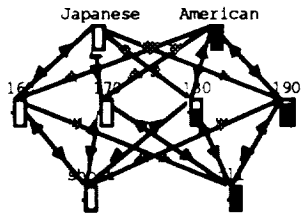


Fig.2.2 CFS representing "tall" American

In contrast, activating only "American", the different distribution from above in middle layer expresses its general meaning in the numeric support set. The propagated activation of "tall" in lowest layer indicates the perception of the height of an American, and it means "(an) American is tall". As we see in this latter example, the meaning of a label in CFS is expressed in multi-layers simultaneously and it is interpreted by each expression.

2.2. Construction of CFS by Learning

We proposed the method to inductively construct CFS as a representation of concepts using neural network learning [10]. It means that the construction is carried out in terms of instances.

Inductive Construction of CFS

CFS are constructed inductively using Hebbian learning. CFS is realized using associative memories in which a link represents a strength of the relation between two concepts. Hebbian learning modifies the strength m_{ij} of links by the product of the activations of two nodes x_i and y_j according to:

$$\dot{m}_{ij} = -m_{ij} + x_i y_j \qquad (3)$$

In this case the correlation matrix is obtained directly from instances such as

"The height of Mark is 175cm. He is tall with a grade 0.8"
"The height of George is 160cm. He is tall with a grade 0.2"

On the other hand CFS are also constructed by the previously proposed algorithm [2] from the fuzzy set.

"tall" = { 0.2/160cm, 0.8/175cm, ... }

generalized from the instances above.

Structural Learning of Concepts

The proposed construction method also covers the structural learning. Since the proposed learning method makes negative correlation for the pairs of elements which are not relating to the concept in question, the obtained CFS does not make unnecessary elements activated. For this reason the proposed method can provide us with a desirable CFS even in support sets which contain verbose

elements.

Composition of Subdivided Knowledge

A complex CFS is realized by composing several pieces of associative memory structured individually. Further composition of pieces of knowledge makes the representation of the concept context dependent. In this procedure the constraints of associative memories are very important.

If C_1, C_2, ..., C_n, denote individual CFSs and M_1, M_2, ..., M_n are their corresponding correlation matrices then we can combine them to obtain a CFS, C, whose correlation matrix, M, is given by:

$$M = M_1 + \cdots + M_n \qquad (4)$$

The following features of CFS allow for solving the shortcomings of purely symbolic knowledge representation paradigms:

1. CFS can represent the context dependent meaning of a concept. At the same time being built through simple combinations it avoids combinatorial explosion.
2. CFS can explicitly represent the concept whose logically explicit representation is impossible.
3. Since CFS can employ a multi-layered distributed structure, many kinds of expressions such as denotative and connotative can be mixed. Inference is performed by passing through layers and propagating activations.
4. As indicated in [11] propagations of activations realize approximate reasoning. Thus, associative memories lend CFS's the characteristics of intellectual information processing such as decrease of fuzziness, bidirectional inference, context dependent reasoning, etc..

3. Fusion of symbolic processing and numerical processing

3.1. Fuzzy Reasoning by means of CFS

As we see above, CFS represent the meaning of a concept in multiple layers. The meaning of the concept is translated into the expression indicated by the distribution of activation in each layer. Since the representation of the meaning in the input layer is translated into a representation in the output layer, the propagation of activation corresponds to reasoning. CFS can realize many kinds of reasoning which behave consistently with other reasoning methods (slight differences are due to different notation).

In particular, rule based approximate reasoning is realized as follows. Consider a rule of the form IF x is A then y is B. A layer consists of nodes representing premises A1,A2, ..., Am, describing x. Another layer consists of nodes representing the consequences B1,B2, ..., Bn, describing y. These layers are connected by a weight matrix M calculated from correspondences of premise Ai and consequence Bj. If the input is x=x*, the concepts A1,A2, ..., Am are activated with the activations being equal to the corresponding membership values of x*. The propagation of activation determined by the activation of the premise layer produces the distribution of activations in the consequence layer, that is B1,B2, ..., Bn. As each activation corresponds to the truth value of each concept, approximate reasoning is realized [12].

As CFS behave beyond the limitation of logic based notation, the following reasoning can be realized using CFS:

1. Propagations which arise from the activation of an abstracted concept show its meaning in the concrete layer. This corresponds to answering the question asking the meaning of the concept.
2. In contrast, the activation of a lower concept determines the activations of an upper concept and it

corresponds to recognition or understanding.

Further, due to its bidirectional features, the reasoning in CFS has various characteristics which cannot be achieved by the logic based paradigm[11].

3.2. Multi-layered Reasoning

Consider a simple example of predicting the currency exchange rate. In the case of a war happening, we use concrete examples from past experience, such as the Gulf War, to predict a precise value. At the same time, we refer the macroscopic knowledge such as "dollar rises in case of emergency" and make rough prediction such that dollar rises up. We can say that the abstracted knowledge described in the upper layer supervises the generous reasoning path and corrects the result of reasoning in the lower layer in terms of concrete knowledge such as numeric data and event data.

In general, quantitative processing or neural network deal with numeric data and are not capable of integrating symbolic semantics. In contrast, symbolic processing suits intellectual information processing, but does not suit numeric processing. Since both processing methods take completely different approaches to knowledge processing and knowledge acquisition, the effective integration of these methods, while desirable, is difficult to achieve in a way of which combines their best features.

A reasoning in a multi-layer structured CFS realizes, to some extent, the integration of these two paradigms. The upper layer is meant to carry out symbolic processing using abstracted concepts while the lower layer to process numeric data and instances. If only the reasoning in the lower layer is used, it gives us precise results, but possibly a wrong reasoning path from macroscopic view point. On the other hand, the reasoning in upper layer alone cannot provide a precise result. Bidirectional association connecting two layers enable us to fuse the simultaneous processing in upper and lower layers to obtain a semantic guide supported by the upper layer and the precise processing supported by lower layer. The correspondences of concepts in upper layer represent the abstracted knowledge and the correspondences of examples or numeric data in the lower layer represent concrete knowledge. Since the concepts in the upper layer are connected with examples in lower layer, these connections result in the fusion of two differently abstracted layers. In the case when more than two layers exist various abstracted processes are carried out at the same time.

The reasoning in a multi-layer structured CFS is carried out according to the following procedure: The activation of the node in premises activates the corresponding several nodes in consequences in the lower layer. At the same time, the result of the semantic information processing in the upper layer propagated by the activation of the node in the premises in lower layer affects the consequences in the lower layer. As a result, the nodes affected by both the direct propagation in the lower layer and the semantic propagation in the upper layer remains to be activated. Finally, a concrete result is obtained in the lower layer and abstracted results are obtained in the upper layer simultaneously. We call **Semantic Guide Line** the supervision of the processing in lower layer by the intellectual information processing in upper layer.

Example 3.1. Decision regarding the amount to steering

When driving a car the amount to steering changes depending on situations. In the case that parking spaces are indicated by a painted line, we usually park the car passing the line. If the spaces are surrounded by borders or walls (as in a garage), another trajectory is considered (to avoid the collision with the wall as in Fig.3.1).

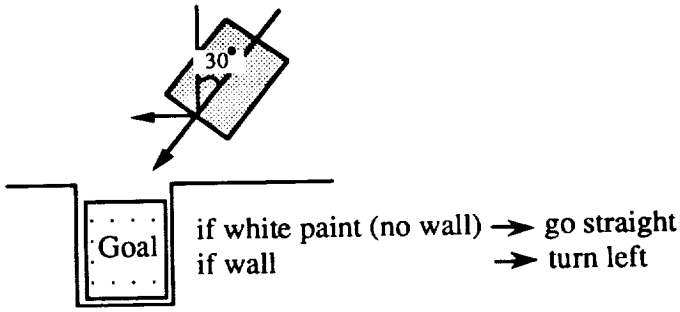


Fig.3.1 Parking Conditions

Consider the case that we decide the amount to steering besides parking space and the direction of the car is placed at 30 degree with the direction of parking space as indicated in Fig 3.2.

We decide the amount to steering using generous rule such as "steer to right to make right turn". The "right" is a concept generalized from various driving experiences and:

1. This kind of symbolic representation is effective to describe explicit and semantic knowledge.
2. However, its indications are vague and can not determine the amount to steering precisely.
3. Its meaning changes depending on the situations such as the position of a car.

On the other hand, cases such as "when the car makes x degree, we steered y degree" are described by concrete numeric values and:

1. The concrete experience indicates the precise amount to steering.
2. However, purely quantitative correspondence of conditions and actions does not suit logical information arising from varieties of conditions.

The CFS fuse both representations consisting of two layers. The lower layer memorizes the correspondences of the numerically described direction of the car and decided amount of steering. Since the lower layer consists of superficial numeric correspondences, it does not recognize the difference between the cases "with wall" and "without wall". In the upper layer, the conditions described by the symbolic notation such as "direction of the car" correspond to the actions such as "with wall" or "without wall". The correspondences of symbols are equivalent to the semantic control rules generalized from experiences . The nodes in the lower layer represent: direction of the car (left nodes) and decided amount of steering (right nodes). The nodes in the upper layer represent: the concept associating with the degree of the car such as "about 45 degree" and "about 90 degree (parallel to the front wall), two nodes on the left, and the conditions "wall" and "no wall" , the remaining two nodes on the left. The nodes on the right side of the upper layer represent the resulted actions such as "Turn left","Go Straight" and "Turn Right". Further, the concepts of the upper layer are connected to the concrete nodes of the lower layer, thus realizing meaning representation.

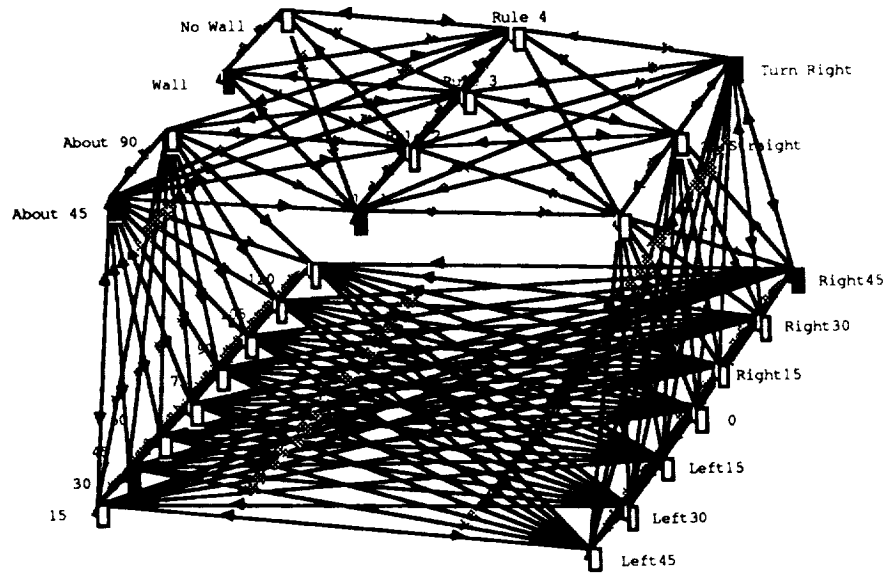


Fig. 3.2 Decision of the amount to steering by two-layered reasoning

Fig.3.2 also shows the conditions and the decided action when the car is placed in 30 degrees with a parking space having a wall. The condition "30 degrees" results in two kinds of actions depending on the cases "with wall" and "without wall". Because the lower layer simply memorizes both actions "15 degree to left" and "45 degree to right" corresponding to the conditions 30 degrees, the correct result cannot be recollected by using only the lower layer.

In the upper layer the recognition of a close wall activates "Turn Right" and it produces the activation of "turn right by 45 degrees" in the lower layer. The results of this multi-layered reasoning are "Turn right" in the upper layer and "turn right by 45 degrees" in the lower layer. This process of determining the actions indicates the successful supervision by the macroscopic views in the upper layer of the lower layer. Moreover, the results of the reasoning are equivalent to the meaning of "right" depending on different conditions.

4. Fusion of top-down and bottom-up processing

Usually natural language processing consists of two steps: (1) parsing and (2) semantic analysis. A lot of meaningless results are obtained by parsing alone. If semantic information could be used simultaneously in the step of parsing it would lead to a more efficient parsing. In image processing, recognition is carried out using characteristic values which are already obtained by low image processing. The fusion of referring a model of an object or the context with the image processing makes the image recognition more efficient. We can say that people simultaneously realize both image processing and recognition.

For the reasons indicated above substantial work has been focused on replacing serial processing by parallel processing [2]. However, this work fails to achieve a real fusion of bottom-up and top-down processing supported by simultaneous information exchange and parallel processing, as it makes use of external procedures (such as for deciding the priority of layers or looping algorithms).

CFS can realize the parallel processing to support the fusion of bottom-up and top-down processing in terms of combining the semantic information processing in upper layer and local processing in lower layer. For example, in image recognition, the upper layer describes the knowledge on a context

while the lower layer describes primitive concepts. The concepts in the upper layer are explained by the primitives in the lower layer. The characteristic values activate the primitives in the lower layer. This results in the activation of the concept in the upper layer. At that time the context described in the upper layer depresses the meaningless patterns of distribution of activation and promotes the meaningful patterns of activations in lower layer. Thus the primitives activated are those affected by the characteristic values and also satisfying the context. This **context sensitive processing** provides us with an accurate result. It uses the context to eliminate vagueness which may come from noisy and vague data and which could otherwise cause misunderstandings.

Example 4.1. Recognition of "THE CAT"

We recognize the words "THE CAT" in Fig. 4.1. Actually the characters in the middle of THE and CAT have exactly the same shape, and the shape can be recognized as either A or H. Therefore if the recognition of the characters is carried out before the recognition of words, it cannot be decided what the character is: A or H. Our actual response recognizing THE CAT indicates the simultaneous processing of character recognition and word recognition (context). CFS can realize this recognition supported by the fusion of bottom-up recognition process and top-down context sensitive processing as in Fig.4.2.

THE CAT

Fig.4.1 THE CAT

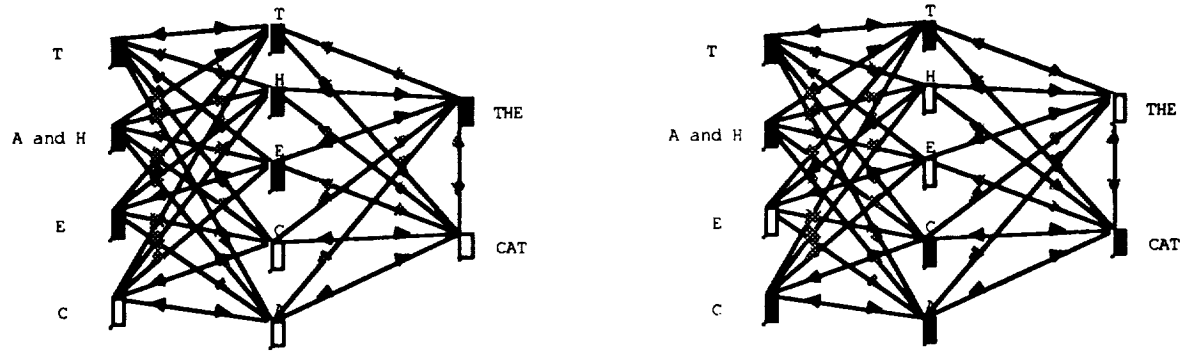


Fig.4.2 The recognition of THE CAT using CFS

The CFS in Fig.4.2 consists of the nodes indicating each character in the lowest layer, alphabets as results of character recognition in the middle layer, and correct words as a context in the upper layer. The lower half of CFS indicates how each character looks like and the upper half indicates the alphabets constructing word. Although the character T, E and C are recognized without vagueness and are connected to corresponding places in the alphabets in the middle layer, the characters of interest which have the shape between A and H are connected to both alphabets to indicate the possibility to be recognized as A or H.

The activation of T, E and the ambiguous character in the lowest layer carry out the recognition. As a result of the propagation of activations, T, H and E are activated in the middle layer and node "THE" is activated in upper layer. The simultaneous recognition indicates that the character in the middle of the word is H and the word is "THE". It should be noticed that context sensitive recognition supported by the upper layer and bottom-up recognition from the lower layer are processed simultaneously.

Example 4.2. Recognition of facial expressions

A facial expression is a vague concept: it is difficult of explicitly describing a facial expression; any descriptions have vague boundaries. In this example, the recognition of facial expression is discussed using multi-layered reasoning by means of CFS. The CFS for facial expressions consists of three layers: the upper layer contains facial expressions, the middle layer contains characteristics of the components of a face and the lower layer contains attributive characteristic values. The facial expressions are described in terms of the following characteristics:the condition of both eyes (UP:upward, HZ:horizontal, DW:downward), and of the mouth (UP, HZ, DW). The above characteristics are described by the following characteristic values: the angle of the edge of both eyes (RA, LA) and the angle of mouth (M) in Fig.4.3. Fig 4.4 shows the object face. The recognition of facial expressions is carried out by activating the node in the lowest layer describing characteristic values.

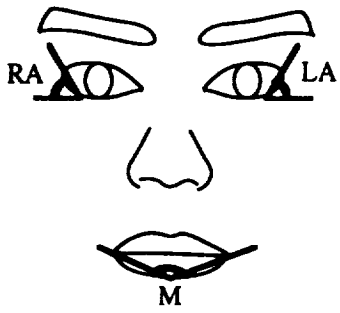


Fig. 4.3 Face characteristic value

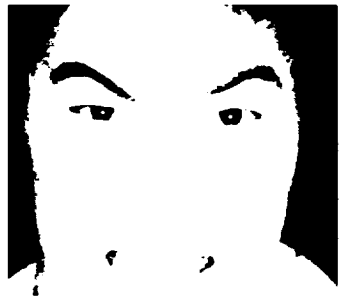


Fig. 4.4 Object image

We can say that humans recognize objects using generous (global) characteristics instead of detecting precise numerical characteristic values. Also, the context constructed by several patterns of facial expressions improves the efficiency and accuracy of recognition. In this section we illustrate the context sensitive image processing by describing general patterns of facial expressions in the middle and upper layers. Fig.4.5 shows the constructed CFS to recognize facial expressions. The general patterns of facial expressions are described by promoting links connecting the characteristics to represent the facial expressions in the middle layer. These patterns are connected to the node in the upper layer standing for facial expressions. The patterns in the middle layer are connected by depressing links. We investigated the recognition using vague characteristic values, which are described by fuzzy sets, to simulate the recognition process by humans without using accurate characteristic values. The object face is recognized as "Angry" and the result is in agreement with our recognition.

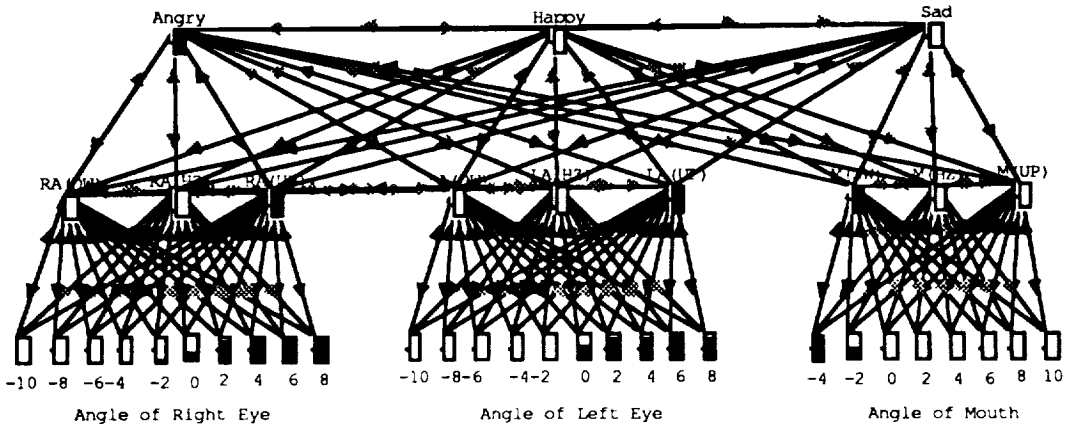


Fig. 4.5 Recognition of facial expressions by means of multi-layered reasoning

In contrast, the recognition using simple logical notation was "Happy" as shown in the following example: facial expression are determined by:

Angry = (Angle of right eye is big) and (Angle of left eye is big) and (Angle of mouth is big)
Happy = (Angle of right eye is small) and (Angle of left eye is small) and (Angle of mouth is small)
Sad = (Angle of right eye is medium) and (Angle of left eye is medium) and (Angle of mouth is big)

Each truth value is calculated as:

Tv(Angry)= min(1.00, 1.00, 0.62) = 0.62
Tv(Happy)= min(0.73, 0.82, 1.00) = 0.73
Tv(Sad) = min(0.92, 0.82, 0.62) = 0.62

Taking the facial expression which has maximum truth value produces the result "Happy".

We also investigated the face recognition of 28 people as shown below and the results show the advantage of context sensitive recognition using CFS.

CFS:	14.3 % fail
logic based:	21.4 % fail

The results show the advantage of context sensitive recognition which is supported by the fusion of bottom-up and top-down processing, in particular, when the recognition starts with error containing vague characteristic values. It also implies the possibility of CFS for image understanding to eliminate the need for precise image processing

5. Conclusion

Fuzzy set theory can be viewed as a stronger and more practical notation than purely symbolic information processing paradigms, connecting the logic based world and the real world. The duality abstract/concrete of the real world is reflected in the intelligent/lack of intelligence duality at the intellectual level. To cope with this duality a knowledge representation paradigm must be able to hierarchically represent both aspects.

Previously we proposed Conceptual Fuzzy sets (CFS) based on the meaning representation of a concept: the meaning of a concept is represented by the distribution of activations of labels naming concepts. In particular, a multi-layer structured CFS represents the meaning of a concept in various expressions in each layer.

In this paper, we proposed Multi-layered Reasoning in CFS. Since the propagation of activations corresponds to reasoning, multi-layer structured CFS can realize multi-layered reasoning which has following features:

1. capable of simultaneous symbolic and quantitative processing (semantic guide line)
2. capable of simultaneous top-down and bottom-up processing (context sensitive processing)

We also showed its effectiveness through illustrative examples.

Acknowledgments

We express our deepest gratitude Dr. Anca Ralescu of LIFE for useful advice. We also would like to express our gratitude to Dr. Hara and Mr. Kobayashi, Science University of Tokyo, for useful advice

and providing us with important data.

References

[1] L.A. Zadeh (1965) : Fuzzy sets, Inform. & Control, Vol. 8, pp. 338-353

[2] R.A.Brooks (1986) : A Robust Layered Control System for a Mobile Robot, IEEE Journal of Robotics and Automation, 2-1

[3] J. Russmussen (1983) : Skills, Rules, and Knowledge: Signals, Signs, and Symbols, and Other Distinctions in Human Performance Models, IEEE Trans. on System, Man and Cybernetics, SMC-13-3, pp.257-266

[4] G.N. Saridis (1983) : Intelligent Robotic Control, IEEE Trans. on Automatic Control, AC-28-5, pp.547-557

[5] T. Sawaragi, T. Norita, T. Takagi (1991) : Fuzzy Theory, Journal of the Robotics Society of Japan, Vol.9 No.2, pp.238-255 (in Japanese)

[6] T. Sawaragi, S. Iwai and O.Katai (1990) : Self-Organization of Conceptual Generalities and Pattern-Directed Learning, Automatica, 26-6, pp.1009-1023

[7] Wittgenstein (1953) : Philosophical Investigations, Basil Blackwell, Oxford

[8] T. Takagi, T. Yamaguchi and M. Sugeno (1991) : Conceptual Fuzzy Sets, International Fuzzy Engineering Symposium'91, PART II, pp. 261-272

[9] B.Kosko (1987) : Adaptive Bidirectional Associative Memories, Applied Optics, Vol.26, No 23, pp. 4947-4960

[10]T. Takagi, A. Imura, H. Ushida and T. Yamaguchi (1992) : Inductive learning of Conceptual Fuzzy Sets, 2nd International Conference on Fuzzy Logic and Neural Networks IIZUKA'92 (to appear)

[11]S. Yamamoto, T. Yamaguchi, T. Takagi (1992) : Fuzzy Associative Inference System and its Features, IEEE International Conference on System Engineering (to appear)

[12]T. Yamaguchi, M. Tanabe, K. Kuriyama and T. Mita (1991): Fuzzy Adaptive Control with An Associative Memory System, IEE CONTROL91 No.332, Vol.2, pp.944-947.

[13]Kobayashi and Hara (1991) : The Recognition of Basic Facial Expressions by Neural Network, International Joint conference on Neural Network, pp 460-466

515-63

Fuzzy Control of an Unmanned Helicopter

ABS. ONLY

M. Sugeno, J. Nishino and H. Miwa
Department of Systems Science
Tokyo Institute of Technology
4259 Nagatsuta. Midori-ku.
Yokohama 227, Japan

P. 1

This paper discusses an application of fuzzy control to an unmanned helicopter. The authors design a fuzzy controller to achieve semi-autonomous flight of a helicopter by giving macroscopic flight commands from the ground.

The fuzzy controller proposed in this study consists of two layers: the upper layer for navigation supervising the lower layer and the lower layer for ordinary rule based control. The performance of the fuzzy controller is evaluated in experiments where an industrial helicopter YAMAHA R-50 is used.

At present an operator can wirelessly control the helicopter through a flight computer with eight commands such as "hover", "fly forward", "turn left", "stop", etc. The results are shown by video.

Fuzzy Logic Mode Switching in Helicopters

Porter D. Sherman and Frank W. Warburton
Advanced Research & Design
Sikorsky Aircraft

The application of fuzzy logic to a wide range of control problems has been gaining momentum internationally, fueled by a concentrated Japanese effort. Advanced Research & Development within the Engineering Department at Sikorsky Aircraft undertook a fuzzy logic research effort designed to evaluate how effective fuzzy logic research effort designed to evaluate how effective fuzzy logic control might be in relation to helicopter operations. The mode switching module in the advanced flight control portion of Sikorsky's motion based simulator was identified as a good candidate problem because it was simple to understand and contained imprecise (fuzzy) decision criteria. The purpose of the switching module is to aid a helicopter pilot in entering and leaving coordinated turns while in flight. The criteria that determine the transitions between modes are imprecise and depend on the varied ranges of three flight conditions (i.e. simulated parameters): Commanded Rate, Duration, and Roll Attitude. The parameters were given fuzzy ranges and used as input variables to a fuzzy rulebase containing the knowledge of mode switching. The fuzzy control program was integrated into a real time interactive helicopter simulation tool. Optimization of the heading hold and turn coordination was accomplished by interactive pilot simulation testing of the handling quality performance of the helicopter dynamic model. The fuzzy logic code satisfied all the requirements of this candidate control problem.

A Neural Based Intelligent Flight Control System for the NASA F-15 Flight Research Aircraft

James M. Urnes
Stephen E. Hoy
Robert N. Ladage
McDonnell Aircraft Company
St. Louis, Missouri

James Stewart
NASA-Dryden Flight Research Facility
Edwards, California

A flight control concept that can identify aircraft stability properties and continually optimize the aircraft flying qualities has been developed by McDonnell Aircraft Company under a contract with the NASA-Dryden Flight Research Facility. This flight concept, termed the Intelligent Flight Control System, utilizes Neural Network technology to identify the host aircraft stability and control properties during flight, and use this information to design on-line the control system feedback gains to provide continuous optimum flight response. This self-repairing capability (Figure 1) can provide high performance flight maneuvering response throughout large flight envelopes, such as needed for the National Aerospace Plane. Moreover, achieving this response early in the vehicle's development schedule will save cost.

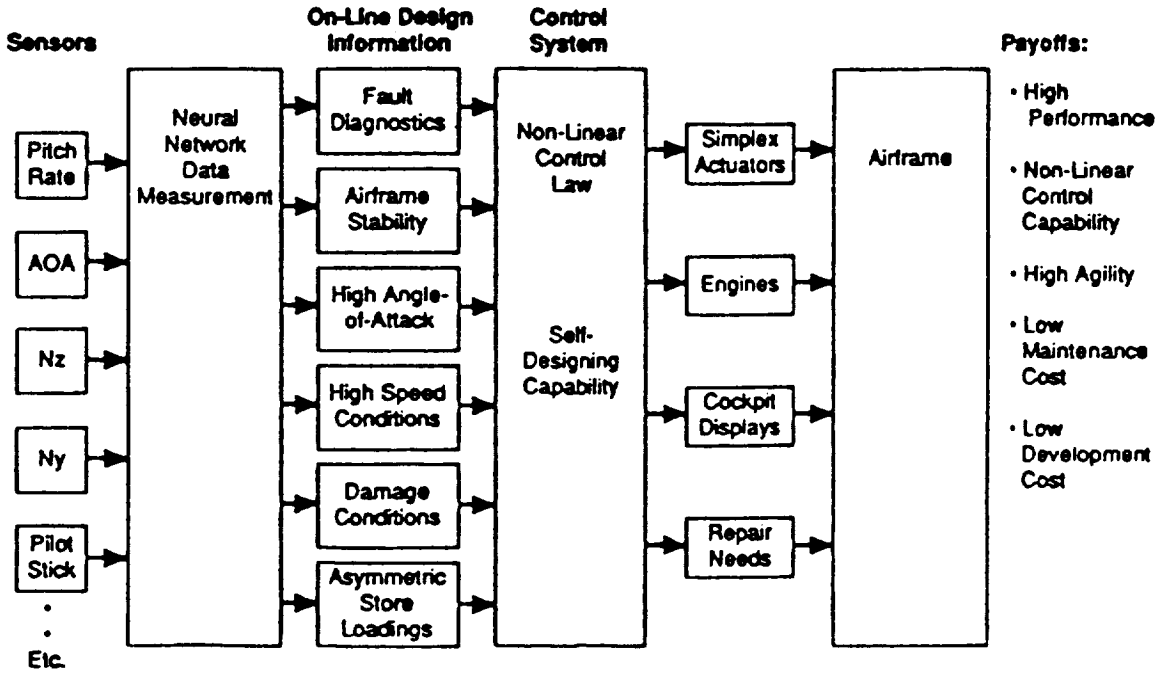


Figure 1. Self Designing Neural Flight System

The Intelligent Flight Control System (Figure 2) incorporates an Aircraft Performance Model to provide the ideal system response. On-time measurements of

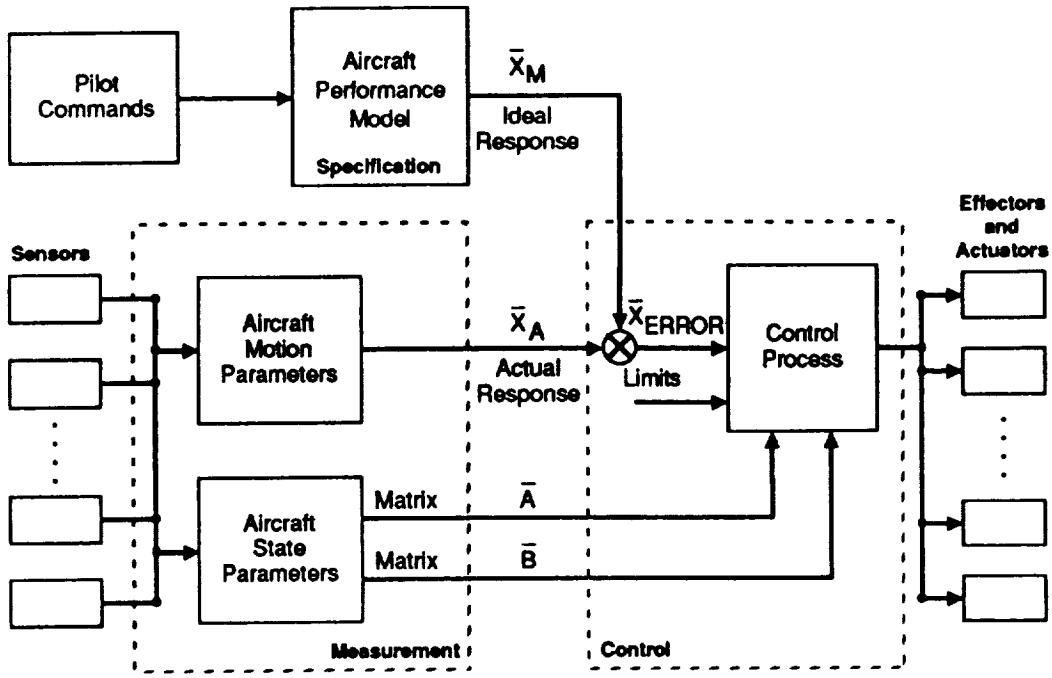


Figure 2. Intelligent Flight Control System

the aircraft state parameters are determined by neural network models that relate aircraft stability coefficients (Figure 3), utilizing aircraft sensors such as Angle of Attack (AOA) as inputs to the networks. Thus, aircraft stability and control coefficients are continuously updated, and used in the control process to achieve the ideal desired response to pilot steering commands. The concept was designed to the NASA F-15

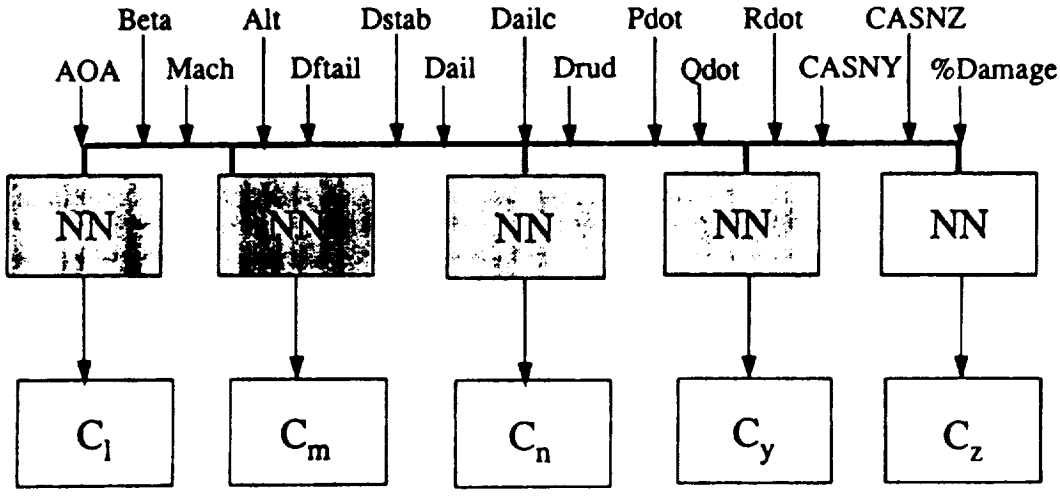


Figure 3. Neural Network Organization

flight vehicle characteristics. Simulated response of the Intelligent Flight Control system to a pilot stick command is shown in Figure 4.

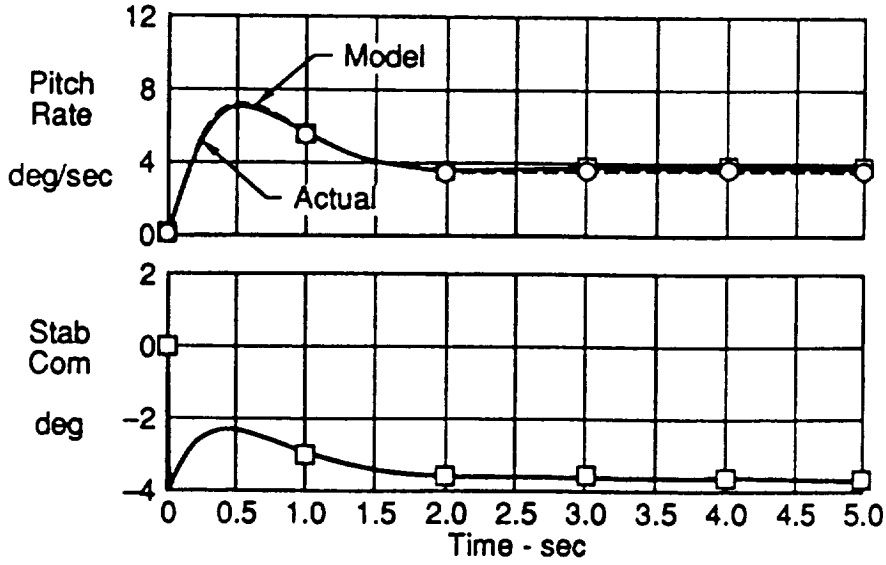


Figure 4. Intelligent Flight Control System Response
Mach 0.7 @ 20,000 Ft, GW=40,685 lb, 1 inch Longitudinal Stick Step

As a test of the concept, aircraft conditions representing a damaged wing was introduced into the problem, using the F-15 wind tunnel data for a 50% missing right wing (Figure 5). Neural Networks were developed to measure the damage, and tested using simulated time histories of the control system sensors as inputs to the networks.

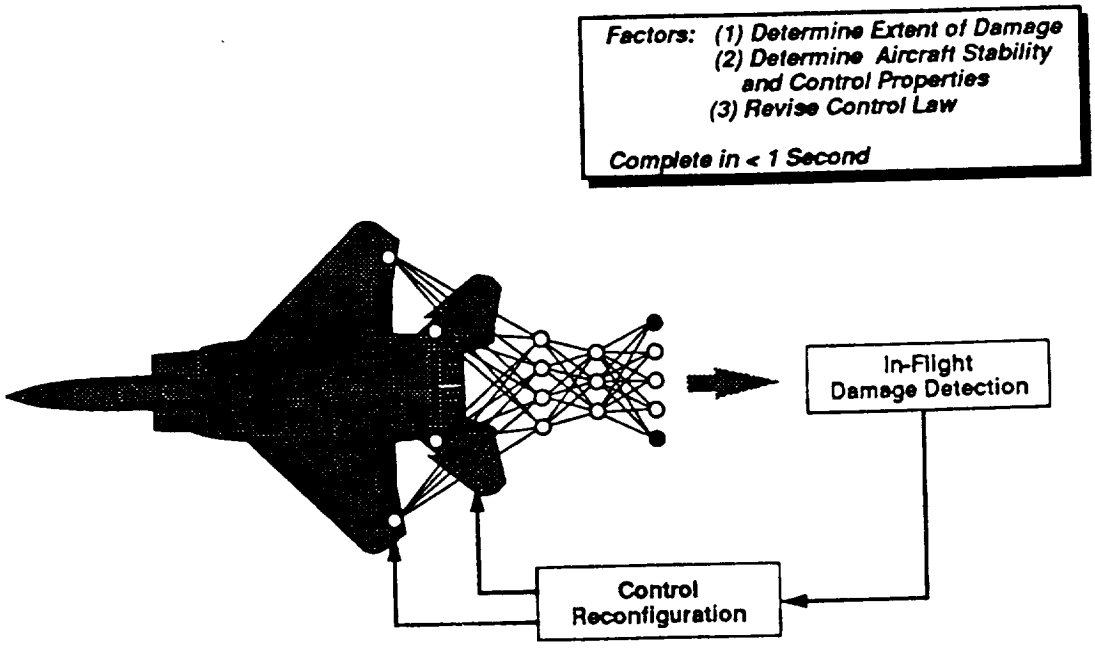


Figure 5. An Example Problem: Control of a Damaged Aircraft

Figure 6 illustrates the aircraft response time history when the wing damage occurs.

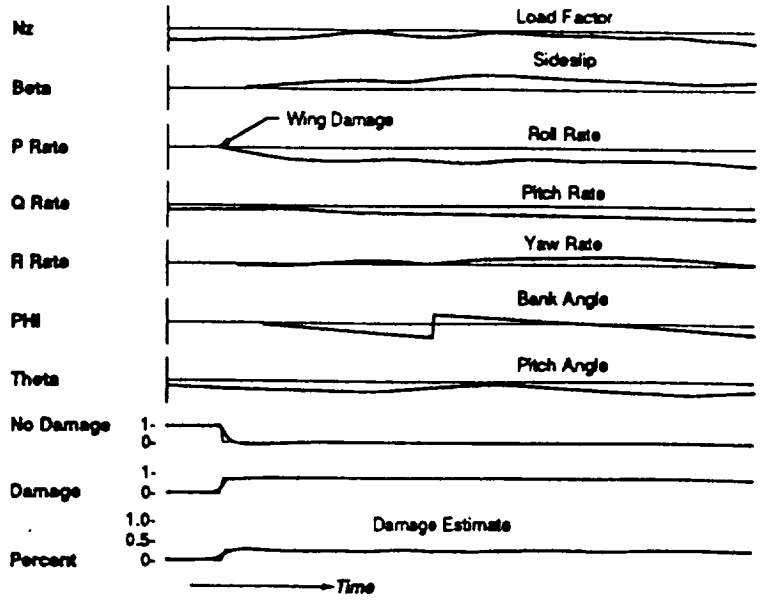


Figure 6. F-15 Response: Right Wing 50% Missing

The information from the Neural Networks will be used to quickly reconfigure the aircraft control surfaces and regain stable, controlled flight.

The Neural-based Self Designing Control Concept that is the basis of the Intelligent Flight Control system can be applied to future fighter and transport vehicles (Figure 7) to optimize engine and flight control performance.

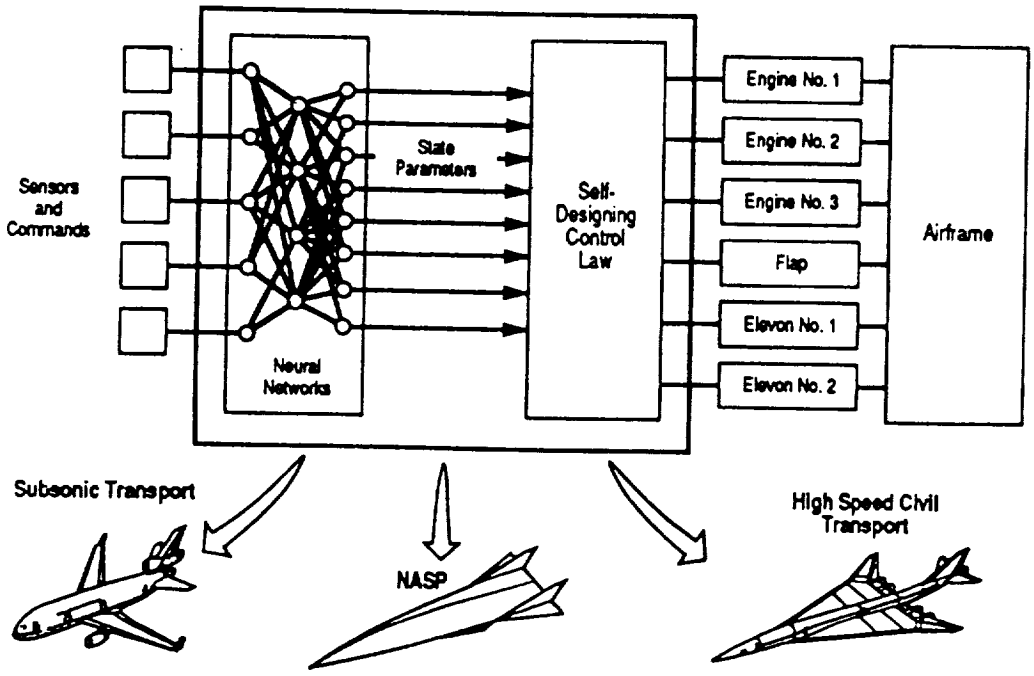


Figure 7. Neural-Based Self-Designing Flight/Propulsion Control

A TELEOPERATED UNMANNED ROTORCRAFT FLIGHT TEST TECHNIQUE

Captain Gregory W. Walker
Arthur E. Phelps III
W. Todd Hodges
NASA Langley Research Center
Aerostructures Directorate, AVSCOM, U.S. Army
Hampton, Virginia 23665-5225

Abstract

NASA and the U.S. Army are jointly developing a teleoperated unmanned rotorcraft research platform at the National Aeronautics and Space Administration (NASA) Langley Research Center. This effort is intended to provide the rotorcraft research community an intermediate step between wind tunnel rotorcraft studies and full scale flight testing. The research vehicle is scaled such that it can be operated in the NASA Langley 14- by 22-Foot Subsonic Tunnel or be flown freely at an outside test range. This paper briefly describes the system's requirements and the techniques used to marry the various technologies present in the system to meet these requirements. The paper also discusses the status of the development effort.

Background and Introduction

Several recent analyses and simulated aerial combat flight tests have demonstrated that agility is a very powerful element of rotorcraft combat survivability. Dynamic stability, maneuverability, and agility are not presently addressed in helicopter wind tunnel testing for both economic and technical reasons, and the investigation of these dynamic issues must therefore be conducted on free-flight vehicles of some type, whether full scale or model scale. Unfortunately, the cost of conducting full-scale flight tests has become so high that it can only be considered for the most important elements of research and development where any other method of test is wholly inadequate. Considerable work is now underway to supplement flight testing with simulation to the maximum extent possible. Simulation, however, can only be exploited when there is a model of the system. Recently developed techniques to validate simulation models require some form of high fidelity flight testing for confirmation. A joint U.S. Army and NASA program is currently underway to evaluate the suitability of using a teleoperated, instrumented, free-flight , reduced-scale powered rotorcraft model equipped with Mach-scaled wind tunnel model rotor systems to refine these validation techniques. This paper provides an overview of the approach and the current status of this free-flight program with an indepth focus on the model's control system.

Free-Flight Research Technique

The free-fight research technique using a model for conducting simulation research is illustrated in figure 1. A specialized flight dynamics research model known as the Free-Flight Rotorcraft Research Vehicle (FFRRV) is flown by a research pilot located in a ground control station. Flight data is telemetered to the ground and recorded in a data acquisition station. The technique of placing the research pilot in the model by means of telepresence technologies rather

*Paper reprinted from IEEE 1992 National Telesystems Conference Proceedings

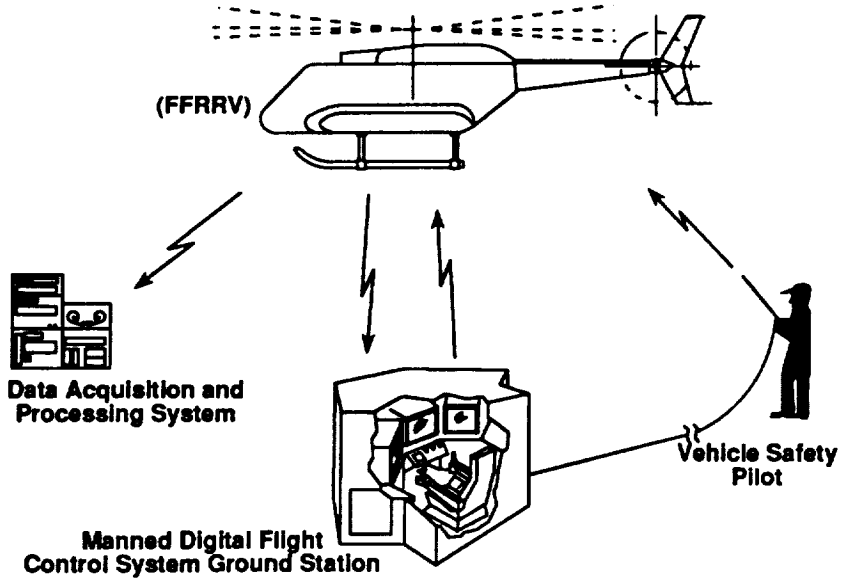


FIGURE 1:
The Proposed Free-Flight Test Technique.

than having him fly by line of sight should ease some of the FFRRV's control systems autonomy requirements because the pilot's perceptions about what is occurring will be keener and his reactions faster. Having the research pilot as an integral part of the aircraft should also allow the pilot to fly more aggressive maneuvers often encountered in nap-of-the-earth (NOE) flight than would be possible with an external pilot. The research pilot's sensory inputs are provided by images from three miniature television cameras and two microphones mounted in the vehicle's nose. The video images are projected onto three, color 26- inch television monitors, and the audio signals are fed into a headset. The video link provides the research pilot sitting in a ground station with a 150 x 35 degree field of view (figure 2). The research pilot's control commands are interrogated by a computer in the ground station and broadcast to the flight vehicle. In addition to the research pilot radio links with the aircraft, there is an external safety pilot who has overall authority over the model in an emergency situation and flies the craft by line of sight like a conventional radio controlled model helicopter.

The Flight Vehicle

The FFRRV is a minimum 225 pound gross weight, aerodynamically scaled model that was designed specifically for conducting flight dynamics research. Almost all of the primary parameters that one would desire to study in rotorcraft research are easily varied. For example, the control system could command excursions in the main rotor RPM to study the resulting variation in dynamics without having to conduct major system redesign and validation as is the case with full scale flight vehicles being flown at an off-design point.

In-house studies indicate that it becomes unfeasible to achieve aeroelastic scaling of a rotorcraft flying in air when the rotor gets any smaller than about 2 meters in diameter. A 2 meter diameter rotor when loaded like a full scale rotorcraft, with 3 to 7 pounds per square foot of disk load, corresponds to a model weight of 200 plus pounds. This rotor size is also scaled similarly to

FIGURE 2:
Ground Control Station Cockpit.

other wind tunnel models that the U.S. Army Aerostructures Directorate operates in the NASA Langley 14- by 22-Foot Subsonic Tunnel.

To maintain the desired flexibility of the test platform there is a core vehicle within the model to which the other essential modules are attached. This core vehicle consists of:

- A steel frame
- 40-horsepower rotary engine and its accessories
- 1.6 KW alternator
- Variable speed ratio belt drive system
- Fixed ratio main rotor transmission
- High speed (greater than 10 inches per second) swashplate actuators
- Flexible shaft and tail rotor drive gearbox

The core vehicle is designed to carry all the loads generated in the system. Tests involving different rotor speeds can be conducted by sizing different diameter pulleys in the belt drive system. Modifying the design rotor speed at this point in the power train greatly reduces costs and the time to modify the system when compared to modifying the rotor speed by using different gear ratios in the transmission. Since the tail rotor is driven off the main drive gearbox with a flexible shaft its location can be moved without requiring a drive system redesign. Attached to this basic core are the additional modules which can be added or modified as the mission requires. The aeroshell itself is one of these additional modules and therefore must only carry the aerodynamic loads that are imposed directly on it. With such an easily modifiable shape some basic phenomena related to detectability or the effects of fuselage shape on agility can be studied quickly and at a very low cost. Some typical configuration studies such as research to obtain a better understanding of unconventional anti-torque systems, like those depicted in figures 3 through 6 could also be

conducted on the FFRRV. The overall effect of this approach is to provide a unique capability to explore new ideas in rotorcraft design in a timely and cost effective way.

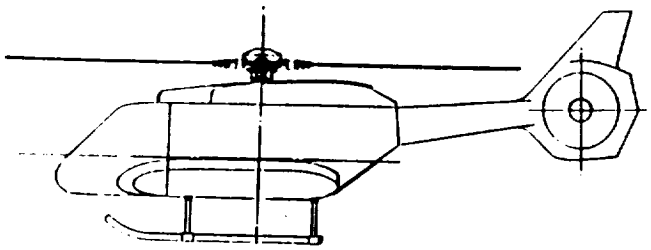


FIGURE 3:
Generic Fenestron Unconventional Anti-Torque System

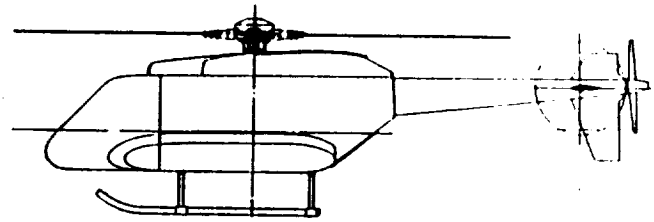


FIGURE 5:
SIKORSKY Swing-Tail Rotor Vectored Anti-Torque System

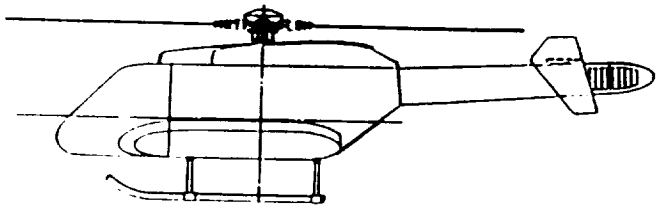


FIGURE 4:
Generic Notar Unconventional Anti-Torque System

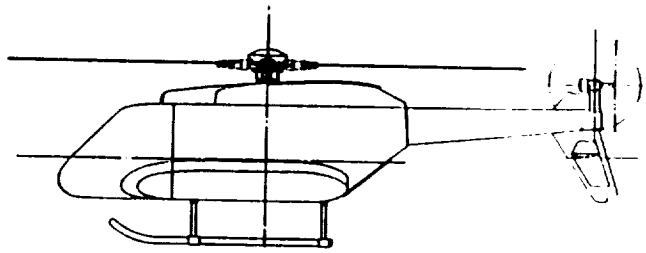


FIGURE 6:
LOCKHEED Pusher Propeller Vectored Anti-Torque System

The Control System

Modularity and flexibility are emphasized in the design of the control system architecture as with all other pieces of the complete system. Subsystem component sets as well as discrete capabilities of the integrated system are broken into separate objects. The objective of breaking the system into submodules facilitates rapid prototyping and testing of new modules and capabilities with minimal impact on existing modules.

The overall goal of the control system is to allow maximum utility to the FFRRV as a research tool by not hampering a test schedule or limiting the scope of a test because of a deficient or inadequate controller for the task. For example, if the researcher requires a certain aggressive flight trajectory to be flown at a certain location over the test range, the desired trajectory could be loaded into the flight computer to fly the vehicle much the same as a human pilot could if he were able to monitor all the parameters of interest quickly enough to maintain them within their test limits. Another desired feature of the control system is to provide a highly stable platform upon which pilot commands can be overlaid. This requirement of the controller is a greater issue with a vehicle of this small scale than it is with a full sized helicopter because the scale factors are different for aerodynamics than for mass and inertia. This difference in scale factors allows the FFRRV to naturally respond quicker to control movements than a full sized helicopter. This "overly sensitive" control responsiveness requires some measure of stability augmentation for piloted flight.

The present control system architecture allows the research pilot to vary the stability and control augmentation system (SCAS) to the specific piloting requirements during flight. The SCAS will operate in various modes in order to achieve this variability. The basic mode is where the control inputs are coupled and an input on one axis has responses on other axes. Another mode is where the controls are uncoupled to a tunable degree where the pilot can vary how much

of an input on one axis affects the off-axes aircraft responses. The most augmented mode is where the vehicle is fully autonomous and the maneuver flown is preplanned. In order to (1) meet these specifications, (2) provide an easily modifiable controller essential for a research tool, and (3) enable some form of vehicle recovery in case of a loss of communication, portions of the control system are located both in the manned ground control station and on the air vehicle. The control systems data analysis and response processing cannot occur entirely on the ground if there is to be any way for the vehicle to sense a loss of communication with the ground station and/or the safety pilot and attempt self-recovery. There are various ways this self-recovery could happen since some of the vehicle's machine intelligence is located on the flight platform and is not entirely on the ground.

A secondary but highly relevant advantage of splitting the control system between the ground and the airvehicle is the potential for reducing the speed and volume of the telemetered data. One computer talking to another in a predefined language can perform at a given level with a lower communication rate than having to encode and decode raw sensor and actuator data at each end of the communication link[1].

The Ground Station Control System: Within the ground station, pilot and researcher commands are processed and broadcast to the flight vehicle for execution. Autonomous flight modes, where the vehicle flies a preprogrammed course on its own, will utilize the ground control station as a source from which to execute the commands. The only autonomous flight planning mode located on the air vehicle is the mode where the vehicle senses a loss of communication and performs a self recovery.

The problem of providing the tunable multilevel controller described above is addressed from both ends of the control authority spectrum. At one extreme the human pilot is in full control without any computer augmentation, and on the other extreme lies an autonomous autopilot capable of flying preprogrammed maneuvers. The middle set of flight modes, where the human augments the autonomous system, is achieved by a blending of the two extremes. In all three modes the resulting commands from the ground station broadcast to the flight vehicle remain the same. Keeping this continuity simplifies design of the airborne controller and places the burden of developing such capabilities on ground based computers where size is not of primary concern. Having this higher level problem solving on the ground eliminates the burden of packing such a capable control system into a volume that will fit into the small airframe of FFRRV.

The Airborne Control System: While looking at the various scenarios which the FFRRV must perform, it quickly becomes apparent that some means of embedding machine intelligence into the flight vehicle would be advantageous. Putting a digital controller on the flight vehicle allows for much faster processing throughput than if all data processing occurred on the ground. Some specific benefits of having a digital controller on the flight vehicle are: (1) Servo control loops require only telemetry to drive a set point. (2) Sensor data can be preprocessed before telemetering it to the high level controller on the ground. (3) It provides the model with some from of machine intelligence that can react to deteriorated communications from the ground.

Being a research tool, where all future uses are not known, it is logical to provide control processing capability on the airvehicle beyond that required in the initial development. This additional capability and speed can be used in two ways:

1. Providing room for growth with new research missions.
2. Allowing rapid testing of unoptimized algorithms without having computational speed become a major limiting factor.

The airborne controller will receive commands from either the safety pilot or the ground station. If the safety pilot commands the vehicle then the airborne controller will ignore any information coming from the ground station and will respond to the safety pilot in a manner similar to a hobby radio controlled helicopter model. If however, the safety pilot has relinquished control to the ground station, as in normal operation, then the airborne controller executes orders from the ground station following a predefined format. This format will be developed to simplify testing the logic of both the airborne and ground station controllers.

The airborne controller will also preprocess the analog signals from the sensor suite and broadcast to the ground control station the following processed sensory information:

- Conditioned sensor data from each sensor.
- A mathematical estimate of the vehicle attitude based on combining the various sensors.

This sensor fusion occurring in the airborne controller relieves the telemetry system from accommodating sensitive analog signals and only requires it to transmit pre-conditioned digital data. This fusion also provides the self recovery capability resident entirely in the airborne controller with accurate knowledge about the vehicle state.

Research Data Recording

The aerodynamic and rotor performance data of interest are collected and transmitted to the ground as a separate entity with minimal interference with other systems on the vehicle. The scope and accuracy of the parameters measured by the data acquisition system mimic that of a wind tunnel Mach scaled rotorcraft model.

The recording of research data occurs independently of the flight data required for the control system. There are two reasons for this:

First, the data of research interest will vary widely depending on the tests being conducted. If the control system data is not a subset of the research data being taken then the additional burden placed on the research data system to acquire the control data will hamper its flexibility. The control systems requirements for data will generally not change whereas the research data collected will vary widely. By separating the two data systems the necessary changes are restricted to one module only.

Second, the control system must be tested and validated irrespectively of the research data or research specific sensors. This allows the vehicle to be developed and flown without any research data collection facility in place. Having this capability facilitates development and makes the system more portable, so it could perform research on various flight test ranges, not just the one it is being developed on.

When a measured parameter necessary for research is the same as one required for the control system, only one instrument which satisfies the more stringent of the two requirements is used to save space. There will, if possible, be two independent pickoff's for the single sensor and all other efforts will be made to isolate any disturbances on one system caused from interrogating the sensor with the other.

If however, the subject of research is related to flight controls, like blade state feedback control, then the control system will require access to the research data recorder. This loop closure, occurring only when necessary, will be on the ground between the Research Data

Acquisition System and the Digital Flight Control System Ground Station to simplify processing and avoid potential contamination of the Airborne Control System.

Status and Plans

We are following a four phase development plan:

1. Proof of concept tests and prototyping of systems.
2. Design and fabrication of a research model.
3. Validation of systems in wind tunnel.
4. Research flight tests.

Currently we are deeply involved in the first two phases of this plan. We are conducting proof of concept flights and control system development with smaller commercial "hobby" helicopters equipped with video cameras, inertial sensors and the associated telemetry (figures 7 and 8). The actual research vehicle is approximately 80 percent complete and has already entered the NASA Langley 14- by 22-Foot Subsonic Tunnel in an unpowered configuration (figure 9). A powerful custom flight computer capable of providing the machine intelligence required on the air vehicle has been designed, built, and is being tested. FFRRV's first flights are scheduled late in the fall of 1992. Prior to these flights the vehicle will again enter the wind tunnel, but this time powered to verify an accurate implementation of the control system. The vehicle will also enter NASA Langley's anechoic chamber for tests to ensure that the assorted telemetry systems supporting the project do not have any transmission dropouts due to antenna blind spots.

The following two sections discuss our current status on the first two phases.

FIGURE 7:
Proof Of Concept Flight Testing Of A Large Commercial Model.

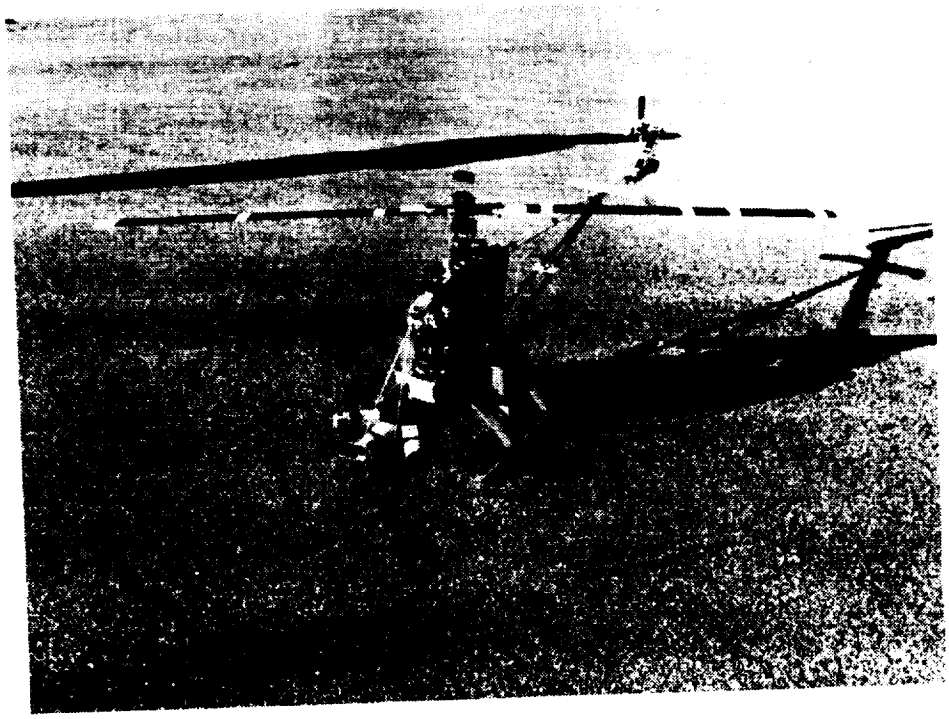


FIGURE 8:
The Large Commercial Model Equipped With Three Video Cameras.

Proof of Concept Tests and Prototyping Efforts

To speed development and reduce the risk of prematurely damaging the research vehicle, we are using commercial hobby-type radio controlled helicopters to resolve issues about systems integration. These RC helicopters are out-of-scale when you look at their aerodynamic surfaces and power systems. However, these models are very useful because we can port much of the integrated systems, debugged on these vehicles, unchanged onto the FFRRV.

Presently we have one model flying at a 200 percent gross weight increase from its original design. Normally the model would have a flying weight of 9.5 pounds, however, the addition of proof of concept equipment, brings the gross weight up to 30 lbs. The benefits of using this model are:

1. The availability of an inexpensive prototype testbed that can fly a large portion of our subsystems for development work.
2. The training of safety pilots on how to recover heavier models. Heavy models respond at different rates than the stock lighter models.

The first use of this heavy out-of-scale vehicle is to clean up the video transmission and receiving system. This work has been going on through the fall of 1991 and is nearing completion.

Following video development, the next task these vehicles will undertake is to fly missions to develop the control system. Initially this effort involves building a mathematical model of the aircraft by performing a system identification of the models and collecting flight data to validate this simulation model. This will be conducted by telemetering sensor data from the aircraft to the

ground. This simulation model of the aircraft will be used to initially tune the control system prior to flight. Once modules of the control system are verified against this simulation model they will be flown and will build upon existing modules that have already gone through this checkout phase, adding incrementally more capability to the model control system. To reduce risk to the research vehicle the control system will only be flown on FFRRV after testing it as much as reasonable on the smaller models.

FIGURE 9:
The Scaled Research Vehicle (FFRRV)
In The NASA 14- by 22-Foot Tunnel.

Design and Fabrication of a Research Model

The Research Flight Vehicle: The initial wind tunnel test of the FFRRV was completed on November 14, 1991. The goals of this test were:

1. Obtain aerodynamic data for baseline studies of the initial fuselage shape.
2. Ensure the tail is adequately sized and placed so it will provide the stability required.
3. Study the effects that forward flight has on the radiator used for engine cooling and ensure there is enough energy being dissipated by the radiator.

The results of this tunnel entry drove slight changes to the initial tail configuration which increased longitudinal and directional stability and provided a capability for in-flight adjustment of pitching moment due to the tail. These changes which involved the addition of vertical tip fins to the ends of the horizontal tail and the incorporation of a short-chord elevator into the horizontal tail surface were verified during the wind tunnel test. The wind tunnel test also identified the need for approximately 30 percent more heat exchange capability to cool the powerplant.

Currently the drive train is being integrated and tuned. We will initially tune the drive train with an electric motor and then later introduce the internal combustion rotary engine. Separating the integration of the drive train and the engine simplifies the tuning required.

A model support system for the wind tunnel has been designed and built which will allow the FFRRV model a limited amount of travel about all three rotational axes and along the vertical axis. This new support system provides a methodological approach to testing the control system in a controlled environment, one motion at a time, prior to flight, and will make possible a new focus in powered rotor testing where body dynamics are the major factor of interest.

The Control System: The distinct tasks that this control system must perform have been logically broken down into separate modules, each with a specific objective (figure 10). The resources necessary to achieve each distinct objective are assigned to the respective module. With this breakdown, parallel development of the separate systems are occurring and will culminate with the final integration and complete system testing.

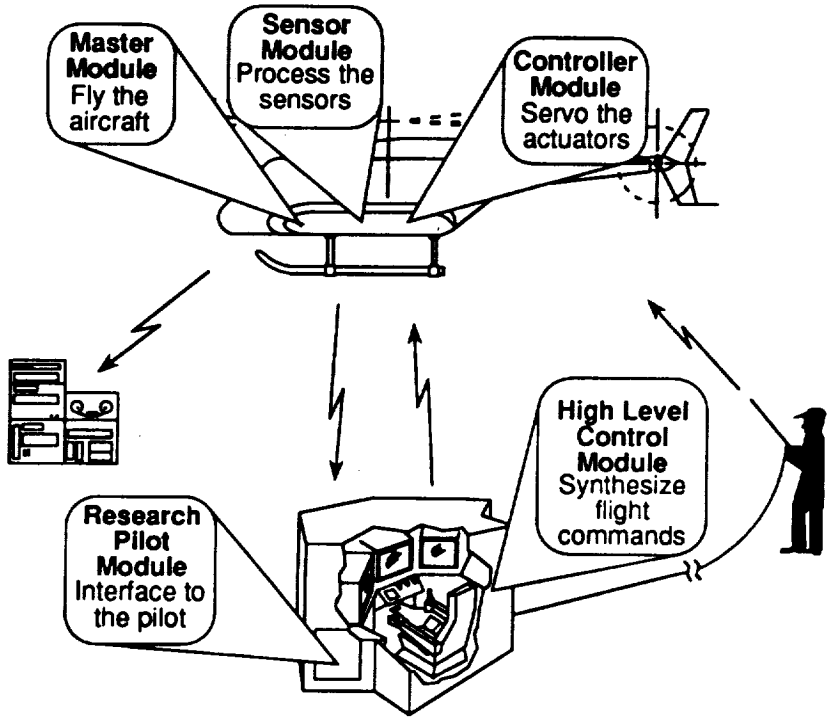


FIGURE 10:
Control System Breakdown.

Two methodologies are presently being compared to determine how best to achieve the three distinct modes of control already discussed: (1) the basic mode, where unfiltered inputs are directly applied to the aircraft (2) the filtered mode, where there is a tunable control augmentation system (3) the autonomous mode, where the aircraft flies a preplanned course. The first methodology under evaluation is based on an accurate model of the aircraft where a nonlinear exact model-following control system, using a model inversion technique, is applied[2]. The second methodology is based on a hybrid of a fuzzy logic controller and a neural network model identifier[3]. At this stage it appears that integrating the human pilot back into the control system will be easier to accomplish using the second approach. Two basic questions presently require resolution: (1) Given the limited information possessed about the model, can a hybrid fuzzy neural controller provide the same precision that an exact model-following controller can? (2) Can an exact model-following controller actually be built with the limited knowledge we have about the model?

The following sections describe the current status of the ground control station and the hardware designed for flight vehicle control system.

Ground Control Station: A working ground station capable of interrogating the research pilot, displaying transmitted video images, and relinquishing control when necessary to the safety pilot is complete (figure 2). Currently a highly modified FUTABA model 1024 9-channel PCM transmitter is operated from the research pilot's seat. In the future, when the ground station is operational with a tunable control system, the FUTABA radio will be replaced with a single high speed telemetry link from a ground computer to an airborne computer. The connection between the safety pilot's radio and the ground station is complete and allows the safety pilot to override control of the model. The video images are each transmitted on their own frequency. The three video receivers are integrated into the ground station enclosure such that the research pilot can tune the video prior to takeoff. Sensory data for the control system is also sent down on a video transmitter.

Initial flights of the heavy weight model helicopter from the ground station are awaiting installation of a stability augmentation system for the aircraft. The RC model, even in its heavy condition, requires stability augmentation prior to flying with cockpit cues without excessive training since it responds so much quicker than full scale rotorcraft.

Airborne Control System: We decided to assign computers with an identical architecture to each submodule in the airborne control system since all the flying modules have identical reliability, weight, and volume restrictions. This decision provides a single development environment and will greatly simplify the final stages of system integration. A market survey of small, powerful computers designed for embedded control application capable of accommodating these specifications was conducted in December 1990. This survey showed that several new 32-bit processors designed for embedded control had just been released. Two microprocessor families of specific interest, the Motorola 683XX and the Intel 80960, had not yet been made into an integrated system small enough to fit into the FFRRV's limited space.

We decided the flight computer must be designed specifically for the mission at hand to maximize its usefulness as a research tool and capitalize on recent microelectronics advances. As a result a control computer based on the Motorola 68332 was developed. The decision to use the 68332 was based on the available software to support it, its advanced internal time processing unit, and because board design is simplified when working with its integrated architecture [4]. The resulting airborne computer system is based on a loosely coupled network of 68332's enhanced with a user selectable amount of:

-Analog input and output for sensor processing
-Additional digital input and output for sensor processing
-Linear Variable Differential Transformer (LVDT) readers for actuator controls
-Flash memory for non-volatile program storage without having to extract the computer from its embedded location in the flight vehicle.
-Large static RAM banks to ease program development, execution, and data collection.

The computer hardware package is very compact measuring 1.5 inches by 4 inches and varies in height from 1 to 5 inches. The height depends on the amount of additional features that a particular module in the multiprocessor control system requires in addition to the basic system.

A multi-tasking real time operating system has been successfully ported to this custom control computer. Low level driver routines, interprocessor communication, and some of the basic I/O functions required in the flight control system have been programmed and tested.

An initial sensor suite was specified and is presently being integrated into the model. The sensor suite is best characterized by its small size and the individual measurements of attitude positions, rates, and accelerations along all 6 axes. Table 1 lists the states being measured and the particular sensor used for observing them [5].

Concluding Remarks

• This is a small scale program which requires a high degree of multi-disciplinary research for its success.

• The program's main goal is to develop a research tool. As the program matures it has a promising future for providing low cost research flight testing where parametric studies can be rapidly executed.

• Successful development of this novel control system will provide a test bed capable of bridging basic artificial intelligence research with systems integration.

• Relatively inexpensive rotor aerodynamic studies will be able to be conducted on hardware in both the wind tunnel and flight completely independent of scale factor corrections.

MEASURED STATES FOR DYNAMIC CONTROL

Quantity	Symbol	Sensor
Range Location	X	Differential GPS
Range Location	Y	" " "
Altitude	Z	" " "
Altitude	Z	15psia Transducer
Near Ground Altitude	Z	Polaroid Transducer
Pitch	Θ	Vertical Gyroscope
Roll	Φ	" " "
Heading	Ψ	Magnetometer
Vertical Rate	w	Variometer
Pitch Rate	q	Reed Rate Sensor
Roll Rate	p	" " "
Yaw Rate	r	" " "
Fwd Acceleration	u'	3 Axis Accelerometer
Side Acceleration	v'	" " "
Vertical Acceleration	w'	" " "
Pitch Acceleration	q'	3 Axis Accelerometer
Roll Acceleration	p'	" " "
Yaw Acceleration	r'	" " "
Air Speed	V	$\alpha\beta$ Bird
Velocity Fwd Angle	α	" " "
Velocity Side Angle	β	" " "
Main Rotor Speed	Ω	Rotary Encoder
Ground Contact Switches	G_C	(4) Micro Switches
Flapping Angle	β	1 Blade Potentiometer

MEASURED STATES FOR ACTUATOR SERVO CONTROL

Quantity	Symbol	Sensor
Main Rotor Collective	Q_C	LVDT and Encoder
Longitudinal Cyclic	A_1	" " "
Lateral Cyclic	B_1	" " "
Tail Rotor Collective	Q_{tr}	Rotary Potentiometer
Throttle Position	S	Rotary Potentiometer

MEASURED STATES FOR MODEL MONITORING

Quantity	Symbol	Sensor
Engine Speed	N_E	Rotary Encoder
Engine Temperature	T_E	Thermocouple
Ambient Temperature	T_{Amb}	" " "
Transmission Temp 1	T_{T1}	" " "
Transmission Temp 2	T_{T2}	" " "
Exhaust Temperature	T_{Ex}	" " "
Clutch Temperature	T_{Cl}	" " "
Water Pressure	P_{Water}	30 psig Transducer
Clutch Actuator Switches	C_{Act}	(2) Micro Switches
Fuel Quantity	F	Float Potentiometer
Lubricant Oil Quantity	L	Float Potentiometer
Alternator Current	I	Ammeter
Alternator Voltage	V	Voltmeter

TABLE 1:
Measured States For Control And Associated Sensors

References

[1] Nelson C. Baker, Douglas C. MacKenzie, Stephen A. Ingalls, "Requisite Intelligence for Producing an Autonomous Aerial Vehicle: A Case Study", Applied Intelligence, The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem Solving Technologies, June 1992.

[2] Michael W. Heiges, "A Helicopter Flight Path Controller Design Via A Nonlinear Transformation Technique," Ph. D Thesis Georgia Institute of Technology, 1989.

[3] Michio Sugeno, Toshiaki Murofushi, Junji Nishino, Hideaki Miwa, "Helicopter Flight Control Based On Fuzzy Logic," presented at the International Fuzzy Engineering Symposium, Yokohama Japan, November 13-15, 1991.

[4] Motorola Semiconductor Technical Data MC68332, Technical Summary 32-Bit Microcontroller, Document BR756/D, 1990.

[5] Anthony Calise, Ken Harrison, Mike Heiges, Robert Michelson, Daniel Schrage, Katherine Taylor, "Analysis For A Stability Augmentation System For A Rotary Wing Target," U.S. Army Final Report No. E-16-619/A-4819-F under contract DAAH01-87-D-0082, November 30 1987.

[6] Willy Albanes, Paul Barker, J.V.R. Prasad "Design Of Fifth-Scale Remote Control Helicopter For Rotor Research," Final Report Contract NAS1-19001, July 20 1990.

Space Time Neural Networks for Tether Operations in Space

Robert N. Lea and James A. Villarreal
National Aeronautics and Space Administration
Lyndon B. Johnson Space Center
Houston, Texas 77058

Yashvant Jani
Togai InfraLogic Inc.
Houston, Texas 77058

Charles Copeland
Loral Space Information Systems
Houston, Texas 77058

Abstract

A space shuttle flight scheduled for 1992 will attempt to prove the feasibility of operating tethered payloads in earth orbit. Due to the interaction between the Earth's magnetic field and current pulsing through the tether, the tethered system may exhibit a circular transverse oscillation referred to as the "skiprope" phenomenon. Effective damping of skiprope motion depends on rapid and accurate detection of skiprope magnitude and phase. Because of non-linear dynamic coupling, the satellite attitude behavior has characteristic oscillations during the skiprope motion. Since the satellite attitude motion has many other perturbations, the relationship between the skiprope parameters and attitude time history is very involved and non-linear. We propose a Space-Time Neural Network implementation for filtering satellite rate gyro data to rapidly detect and predict skiprope magnitude and phase. Training and testing of the skiprope detection system will be performed using a validated Orbital Operations Simulator and Space-Time Neural Network software developed in the Software Technology Branch at NASA's Lyndon B. Johnson Space Center.

1.0 Introduction

NASA and the Italian Space Agency plan to fly the Tethered Satellite System (TSS) aboard the Space Shuttle Atlantis in July, 1992. The mission, lasting approximately 40 hours, will deploy a 500 kg satellite upward (away from the earth) [1,2] to a length of 20 km, perform scientific experiments while on-station, and retrieve the satellite safely. Throughout the deployment, experimentation, and retrieval, the satellite will remain attached to Atlantis by a thin tether through which current passes, providing power to experiments on-board the satellite. In addition to the scientific experiments on-board the satellite, the dynamics of the tethered satellite will be studied. The TSS dynamics are complex and non-linear due to the mass of the tethered system and the spring-like characteristics of the tether. A high fidelity finite element model of the TSS, in which the tether is modelled as a series of beads connected via springs (Fig. 1), realistically represents the dynamics of the TSS, including the longitudinal, librational, and transverse circular oscillations referred to as "skiprope" motion. Since the satellite is a 6 degree of freedom vehicle, it also properly exhibits the satellite attitude oscillations. The skiprope motion is generally induced when current pulsing through the tether interacts with the Earth's magnetic field [3, 4]. The center bead

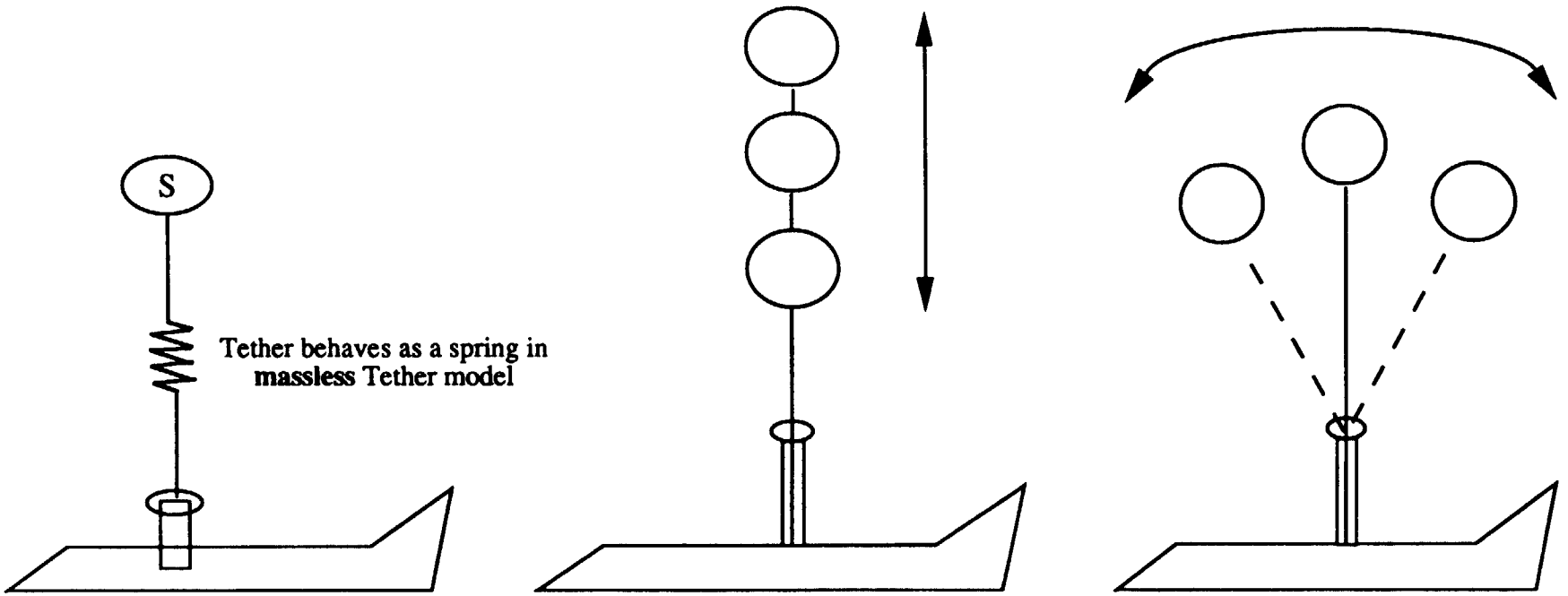


Fig. 1-a Longitudinal and Librational Oscillations in a Tethered Satellite System using Massless Tether Model

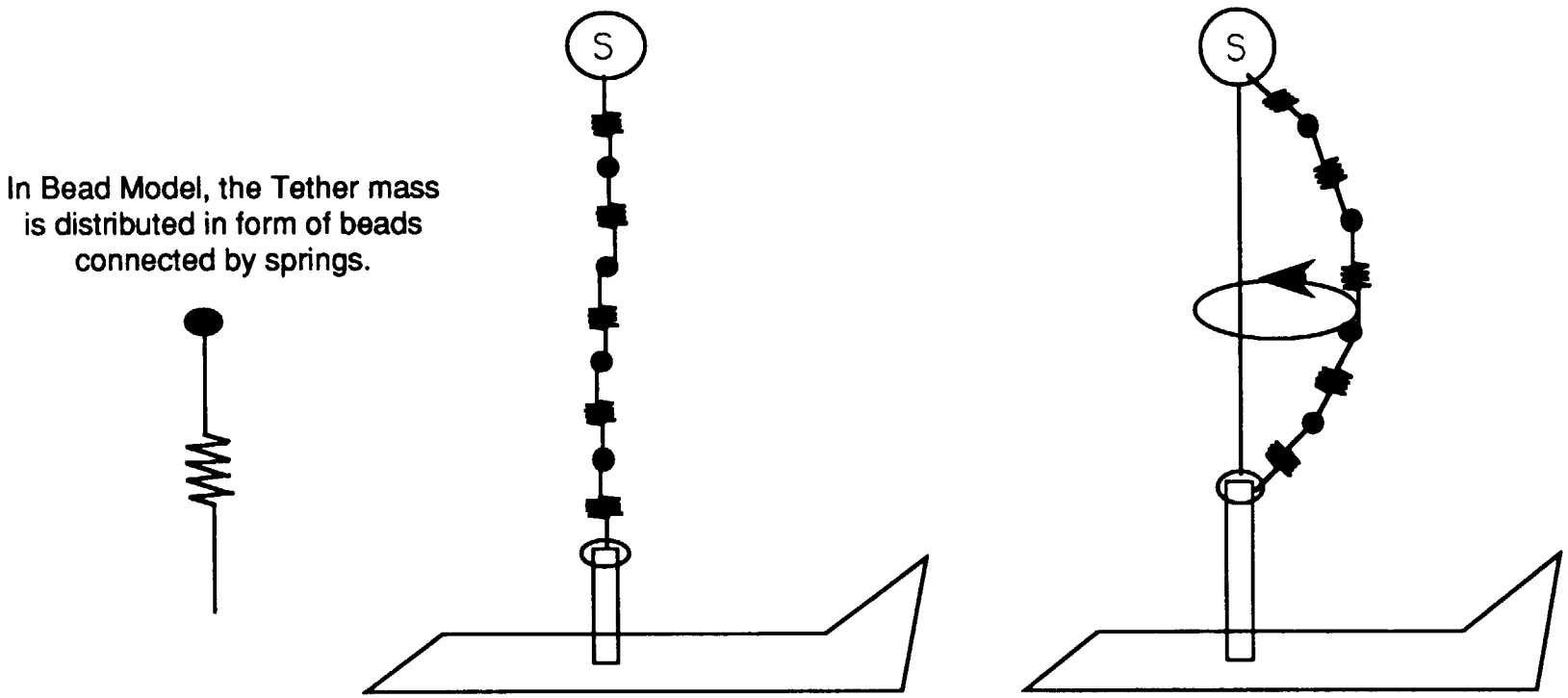


Fig. 1-b In Tether Bead Model, Skiprope Effect or Transverse Oscillations are created when Current interacts with Geomagnetic Field

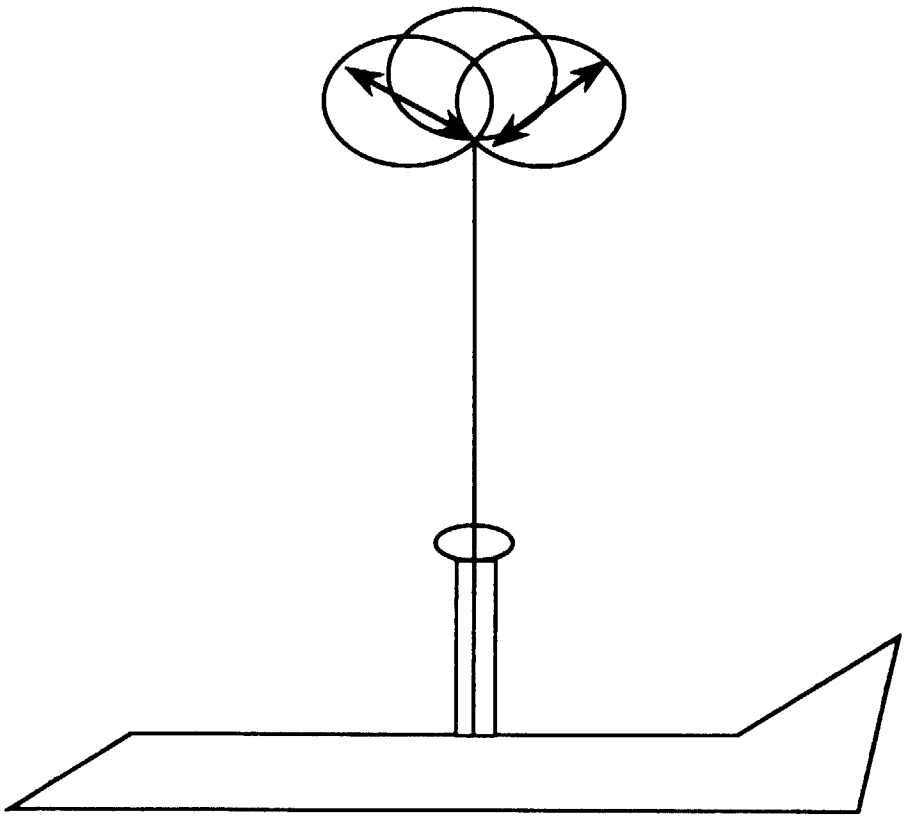


Fig. 1-c Since Satellite is a 6 Degree Of Freedom vehicle, it exhibits Attitude Oscillations which are coupled with the Skiprope Effect

typically displaces the most from the center line. Thus, the "skiprope" can be viewed by plotting a trajectory of the mid-point of the tether as it is retrieved slowly from the on-station-2 phase in high fidelity simulation test cases. As shown in Fig. 2-a, the circular skiprope motion is very simple when there are no perturbing forces. However, when the current is partially flowing, or the current is pulsing with the satellite spin, the skiprope motion is very non-linear as shown in Fig. 2-b and 2-c. Detection and control of the various tether modes, including the "skiprope" effect, is essential for a successful mission. Since there are no sensors that can directly provide a measure of skiprope oscillations, indirect methods such as the Time Domain Skiprope Observer [4] and Frequency Domain Skiprope Observer [3] are being developed for the TSS-1 mission. We are investigating a Space-Time Neural Network (STNN) based skiprope observer.

The Software Technology Branch (STB) is evaluating technologies such as fuzzy logic [5], neural networks [6,7], and genetic algorithms for possible application to various control and decision making processes [8,9,10] for use in NASA's engineering environments. This paper describes the feasibility of applying neural networks, in particular Space Time Neural Network (STNN), to detect and possibly control the skiprope phenomenon using training data from real–time man–in–the–loop simulations. The first phase, detection of skiprope effect (in terms of magnitude and phase angle with respect to the tether line), is vital for tether dynamics control. An STNN architecture has been developed which provides the capability to correlate the time behavior and generate the appropriate output parameters to identify and control skip rope behavior. In this paper, a brief description of the STNN architecture is provided (section 2) along with a scenario of the TSS mission with a focus on the 'skiprope' effect (section 3). The STNN configuration used in our initial test cases and preliminary results are described in section 4. Advantages of utilizing STNN over conventional methods for the detection of skiprope parameters are discussed in section 5. A summary including future activities is provided in section 6.

2.0 Space Time Neural Networks

The Space-Time Neural Network [11] is basically an extension to a standard backpropagation network in which the single interconnection weight between two processing elements is replaced with a number of Finite Impulse Response (FIR) filters. The use of adaptable, adjustable filters as interconnection weights provides a distributed temporal memory that facilitates the recognition of temporal sequences inherent in a complex dynamic system such as the TSS. As shown in Fig. 3a, the inputs are processed through the filters before they are summed at the summing junction.

Instead of a single synaptic weight with which the standard backpropagation neural network represents the association between two individual processing elements, there are now several weights representing not only spatial association, but also temporal dependencies. In this case, the synaptic weights are the coefficients to adaptable digital filters:

$$y(n) = \sum_{k=0}^{N} b_k x(n-k) + \sum_{m=1}^{M} a_m y(n-m) \qquad (1)$$

Here the x and y sequences are the input and output of the filter and the a_m's and b_k's are the coefficients of the filter.

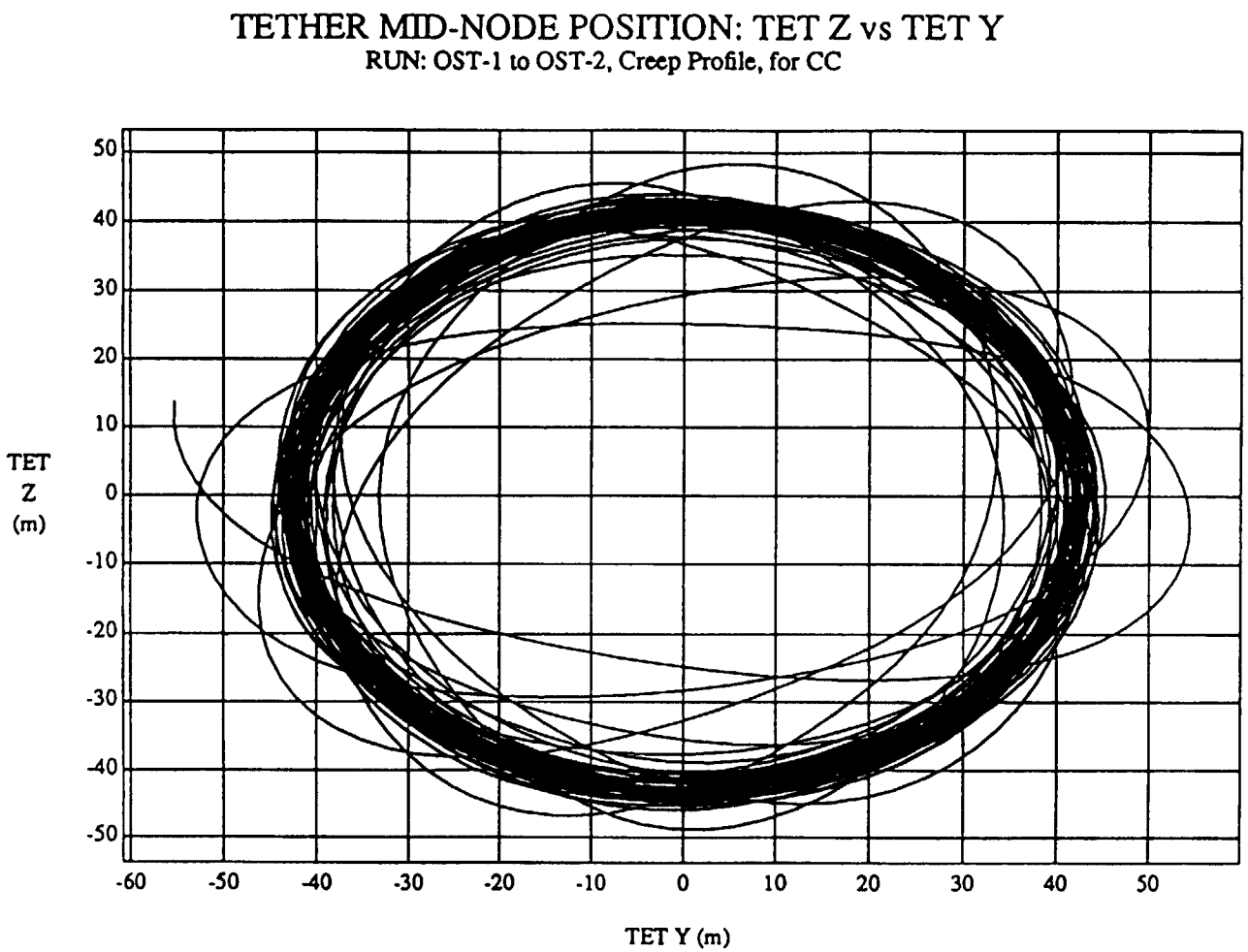


Fig. 2-a Circular Skiprope Motion

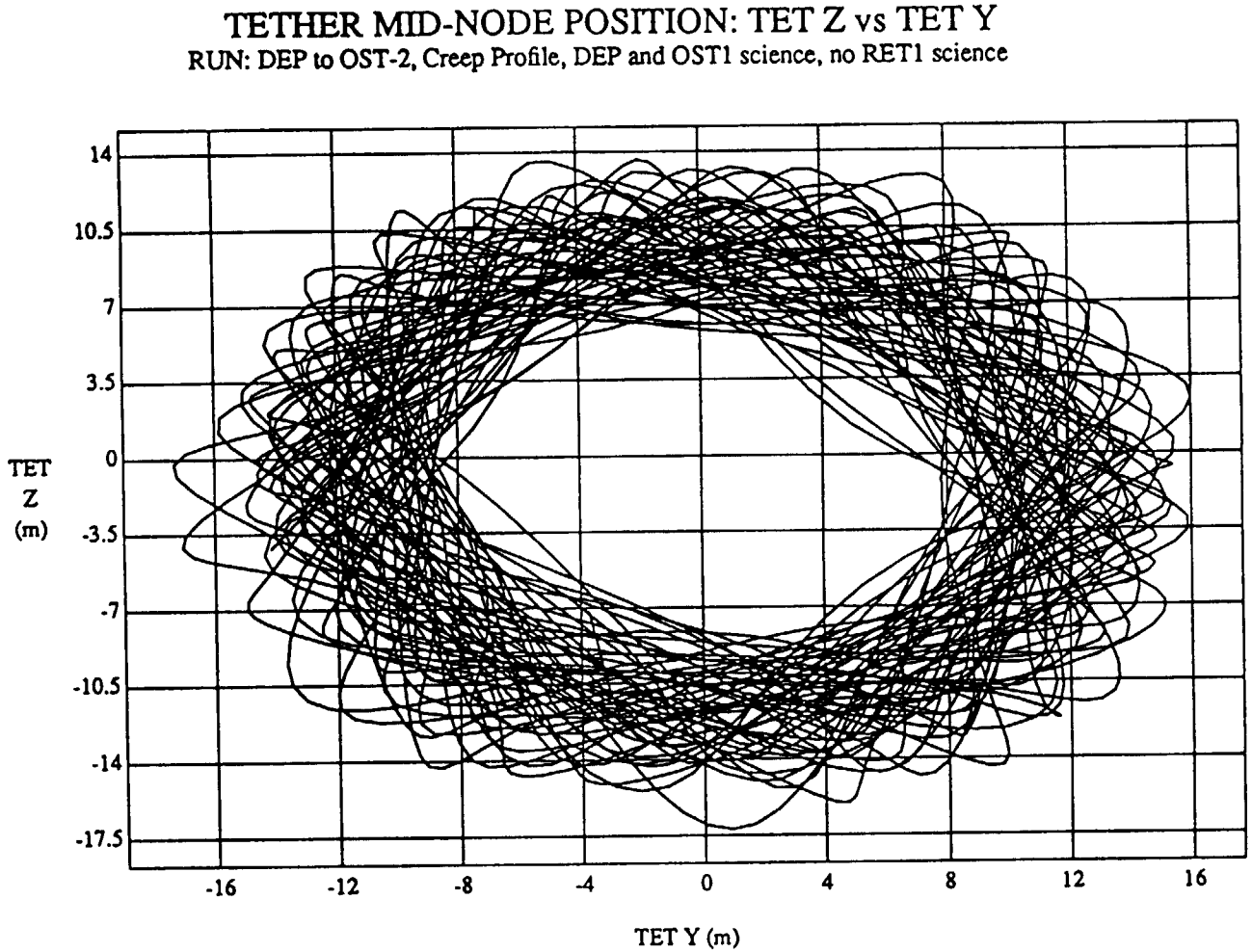


Fig. 2-b Skiprope Motion Resulting from Partial Current Flow

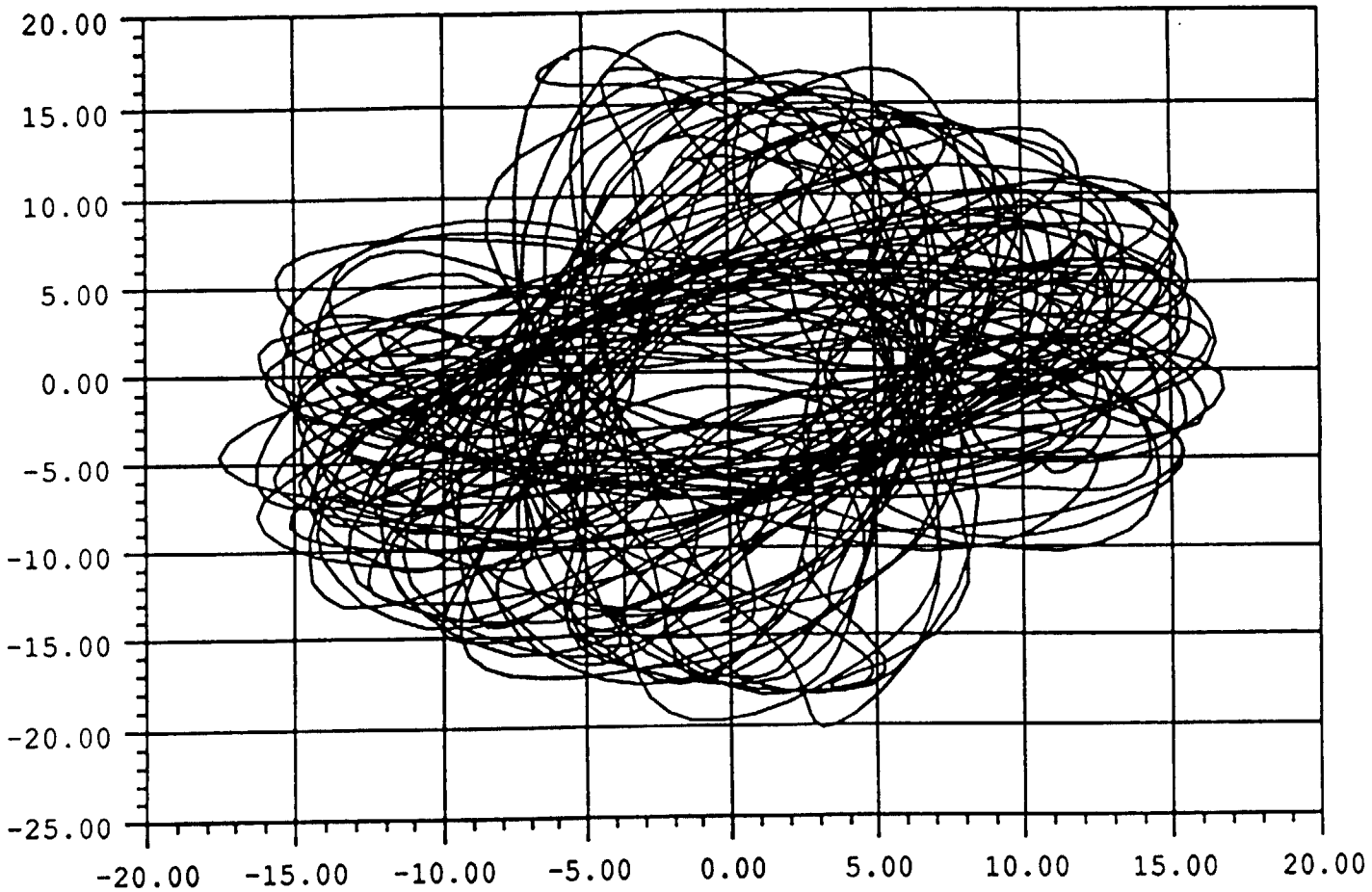


Fig. 2-c Skiprope Motion Resulting from Current Pulsing and Satellite Spin

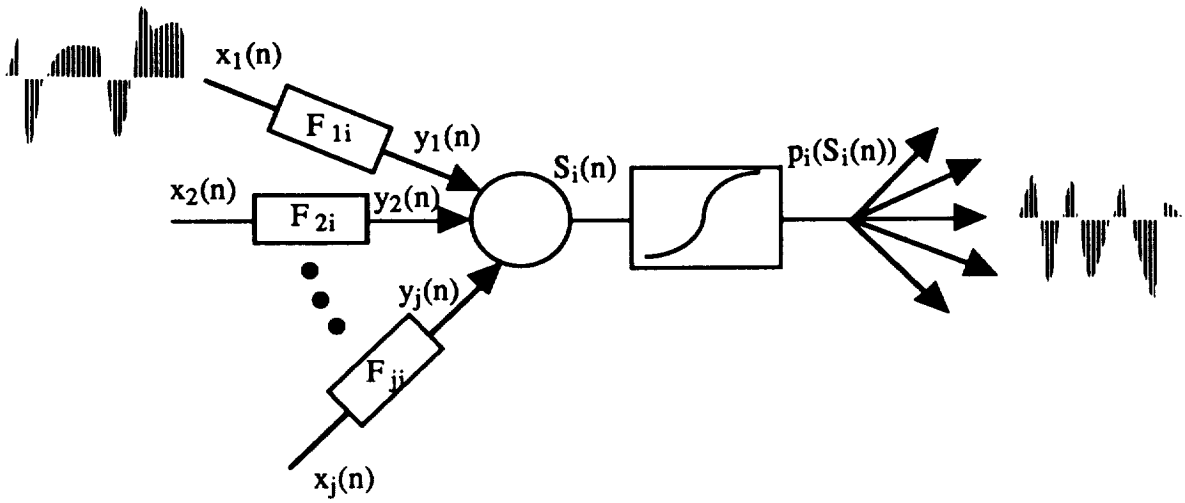


Figure 3 a.... A pictorial representation of the Space-Time processing element.

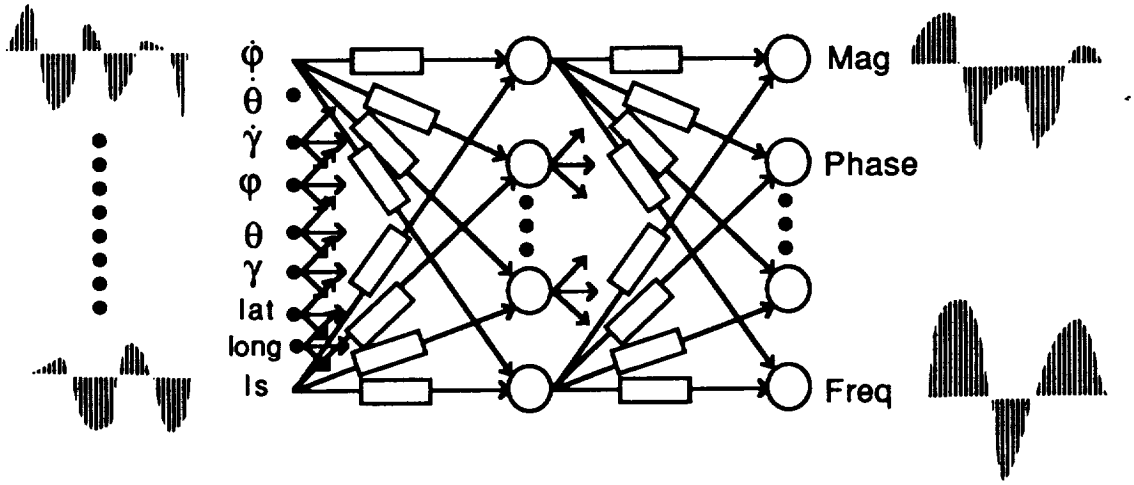


Figure 3b - A depiction of a STNN architecture showing the distribution of complex signals in the input space.

A space-time neural network includes at least two layers of filter elements fully interconnected and buffered by sigmoid transfer nodes at the intermediate and output layers. A sigmoid transfer function is not used at the input. Forward propagation involves presenting a separate sequence dependent vector to each input, and propagating those signals throughout the intermediate layers until the signal reaches the output processing elements. In adjusting the weighting structure to minimize the error for static networks, such as the standard backpropagation, the solution is straightforward. However, adjusting the weighting structure in a recurrent network is more complex because not only must present contributions be accounted for but contributions from past history must also be considered. Therefore, the problem is that of specifying the appropriate error signal at each time and thereby the appropriate weight adjustment of each coefficient governing past histories to influence the present set of responses. A detailed discussion of the algorithm can be found in the provided reference [11]. For the tether skiprope detection, the parameters like satellite spin in terms of roll, pitch and yaw body rates, angles which are derived from these rates, and length and tension will be input, while, the skiprope magnitude and phase will be the output of the net as shown in Fig. 3b.

3.0 Tether Skiprope Phenomenon in Space Operations

The TSS mission is divided into five phases: Deployment, On-station 1 (OST1), Retrieval to a 2.4 km. length, On-station 2 (OST2), and Final Retrieval. The tether motion exhibits longitudinal as well as librational modes as shown in Fig. 1 due to the interaction between gravity gradient forces and spring like characteristics of the tether. These natural modes are damped by controlling the deployed length and length rate using the reel motor drive. A conventional controller is baselined to utilize the sensed length and length rate measurements from sensors. Performance of this baseline controller is adequate in controlling these modes during all phases.

During the OST1 phase, scientific experiments planned include pulsing large electric currents through the conducting tether. Because of interaction between the Earth's geomagnetic field and the pulsing current, transverse circular oscillations known as the 'skiprope' effect as shown in Fig.2 are induced in the tether motion. Simulation results with a 19 bead model of the tether showed the skiprope magnitude between 20 and 70 meters at 20 km. tether length during the OST1 phase of the mission. The skiprope motion is slightly elliptical, i.e. asymmetric around the axis defined by orbiter-satellite line and thus the determination of phase angle becomes involved. To visualize the skiprope motion, we have plotted the z-y motion of the central bead as shown in Fig. 2-a. This motion is the departure from the line that connects the satellite and orbiter. The skiprope motion is very regular for a simple case with no satellite spin, and no current pulsing. When the satellite has spinning motion and the scientific experiments pulse the current through the tether, the skiprope motion is very non-linear as shown in Fig. 2-c. Various combinations of current flow and satellite spin can result in a motion similar to Fig. 2-b. For our initial study, we have utilized the skiprope motion from a simple case. In later test we progressed to more complicated skiprope motions.

Simulation results have shown that the librational amplitude increases about 6 times if there is a skiprope motion present during the retrieval. If librational amplitude is above a critical value, then, the librational oscillations must be damped to a safe value using the Orbiter pitch jets before any further retrieval can be performed. The skiprope amplitude remains between 10-20 meters during the OST2 phase. If the skiprope motion is not properly damped at 2.4 km. then two issues arise during the final retrieval: 1) The satellite pendulus

motion increases significantly (about 6 degrees per meter of skiprope amplitude) such that the attitude control for the satellite fails. 2) The departure angle of the tether from the boom tip may go beyond 60 degrees, thus causing concerns about the tether hitting the Orbiter tail and getting tangled. This may result in a mission failure due to a situation known as "wrap-around" where the safety of the crew and the orbiter is questionable. Therefore the control of skiprope magnitude is very important during the final retrieval phase.

The satellite attitude motion depends on the orbital environment (e.g. perturbations from aero torques) as well as the tension resulting from tether modes. The longitudinal and librational modes affect the satellite rates because of tension coupling at the attach point. However, simulation results indicated that the skiprope effect induces highly characteristic oscillations in the satellite attitude motion (Fig. 1). Due to the dynamical coupling, the skiprope energy seems to be transferred to satellite attitude oscillations. Currently there is no direct measurement available that can provide information regarding the skiprope motion, particularly, the magnitude and phase of the skiprope. Since the satellite attitude behavior is coupled with skiprope, it is possible to utilize the satellite rate (and angles derived from them) information to detect the skiprope parameters.

Controlling the skiprope effect requires knowledge of the magnitude and phase angle of the tether. The amount of pitch torque applied using the Orbiter pitch jets is proportional to the skip rope magnitude. To decrease the skiprope magnitude, the pitch jets are used when the phase angle is 0 or 180 degrees. A pitch pulse increases the skiprope magnitude, if the phase is 90 or 270 degrees. Thus, the phase angle provides the timing of pitch pulse, while the magnitude establishes the amount of pitch torque to be applied.

Performance of the Neural Network Skiprope Observer (NNSO) will be evaluated in terms of the following top level requirements for the Time Domain Skiprope Observer (TDSO).

1.) Operate during all mission phases, where length < 1000 m.
2.) Operate during satellite spin.
3.) Operate during current flow.
4.) Operate during satellite spin and current flow.

In addition to these general requirements, the following goals should be met.

1.) During periods in which the skiprope motion is circular, and there exists no current flow and no satellite spin, the observer must predict skiprope amplitude to within 10% of actual amplitude or 5.0 m, whichever is greater, and predict phase to within 10 degrees

2.) During periods in which the skiprope is circular or non-circular, and there exists current flow and satellite spin, and after 20 minutes settling time, the observer must predict amplitude to within 20%, and phase to within 45 degrees.

3.) The observer must predict in-plane and out-of-plane libration to within 1 degree of actual values.

4.0 STNN configurations and Test Results

To provide data for the STNN training and testing, we have logged data from a high fidelity simulation of the TSS-1 mission, including the OST2 phase. The purpose of OST2 is to halt the retrieval phase at 2.4 kilometers so that skiprope motions and librational oscillations can be reduced to safe magnitudes to allow for final retrieval. Several different

simulation runs were used to gather data for STNN training. The simulation runs are consistent with the requirement that the skiprope observer must be capable of performing during various combinations of current flow through the tether, and satellite spin. For example, our first set of test cases are based on data from a simulation in which there is no current flow or satellite spin which results in a circular skiprope motion. Another simulation represents the case in which current flows through the tether only during the on-station phase, and the satellite is in yaw-hold. A third simulation represents continuous current flow, and satellite spin at 4.2 degrees/second. These three scenarios will form the basis for STNN skiprope observer training and testing, and are consistent with simulations used for testing the Time-Domain Skiprope Observer (TDSO)[4] which will be used for skiprope recognition during TSS-1.

In each simulation run, we have logged 3,000 to 4,000 data points just prior to and during OST2 phase for neural network training. In our initial test cases we use roll rate, pitch rate, roll and pitch position, tether tension, and tether length as inputs to the neural network. Based on these inputs, we hope to find a neural network configuration which will predict skiprope amplitude and phase. The assumption that satellite rates are coupled with skiprope motion is consistent with the baseline Time Domain Skiprope Observer (TDSO) which will be utilized during the TSS-1 mission. The following sections discuss the results of four major test cases.

4.1 Identifying Skiprope Amplitude

To determine the feasibility of using STNN for skiprope detection, we initially trained on data from a simple, circular skiprope case with no satellite spin or current flow through the tether, which is consistent with the first requirement listed above. The data used for training and testing in preliminary tests reflect a near circular skiprope , as depicted in Fig. 2-a. Future test cases will concentrate on more difficult skiprope conditions, such as that pictured in Fig. 2-c, which results from satellite spin and current flow through the tether. The test cases described in this section attempt to evaluate the STNN's ability to identify skiprope amplitude only. We will present the results of test cases involving other skiprope parameters in subsequent sections.

The STNN configuration used in our initial test cases has six inputs, one output, 30 hidden units and 40 zeros for the filters. The choice of the inputs is based on the coupling between the satellite attitude and rates. We have used the roll and pitch angles, roll and pitch rates, tension and deployed length as input to the STNN, and skiprope amplitude as the output. Yaw angles and rates were not used on the inputs in this case because the satellite remains in yaw hold throughout the simulation. To determine if the network is capable of learning the training data, we first train and test on all available (4,001 in this run) I/O pairs. Fig. 4-a shows that the STNN reaches a MAX error of 0.07, and an RMS error of 0.02 within 150 cycles of training. As shown in Fig. 4-b, the STNN predicts skiprope amplitude to within about 3 meters of actual amplitude. For clarity, we have shown STNN performance on I/O pairs 1501-2000. This is fairly representative of the STNN's performance on all 4,001 I/O pairs.

In the next test case, in order to evaluate the network's ability to recognize previously unseen data based on limited exposure to training data, we train on only the first and last 200 I/O pairs, and test on the middle 3,600. Neural network practitioners typically test a network by training until the network reaches some minimum error, and then presenting the test data to the network. In our tests we alternate training and testing throughout a number of training cycles so we can see the correlation between training cycles and test errors. For this reason, our test error plots indicate errors over several presentations of the test data, rather a single presentation. Fig. 4-c shows the errors produced upon presentation of the

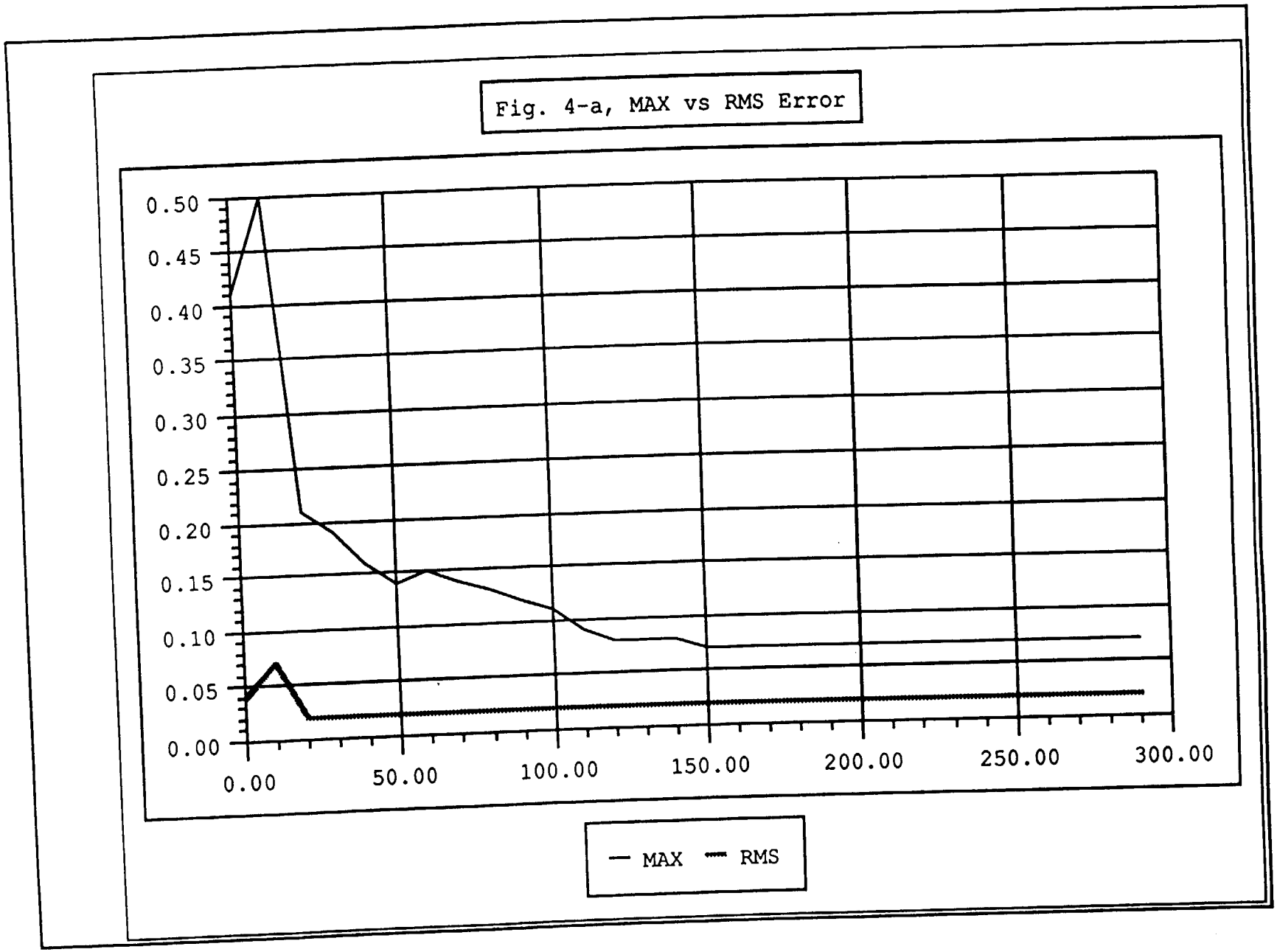


Fig. 4-a, MAX vs RMS Error
0.50
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0.00
50.00
100.00
150.00
200.00
250.00
300.00
MAX
RMS

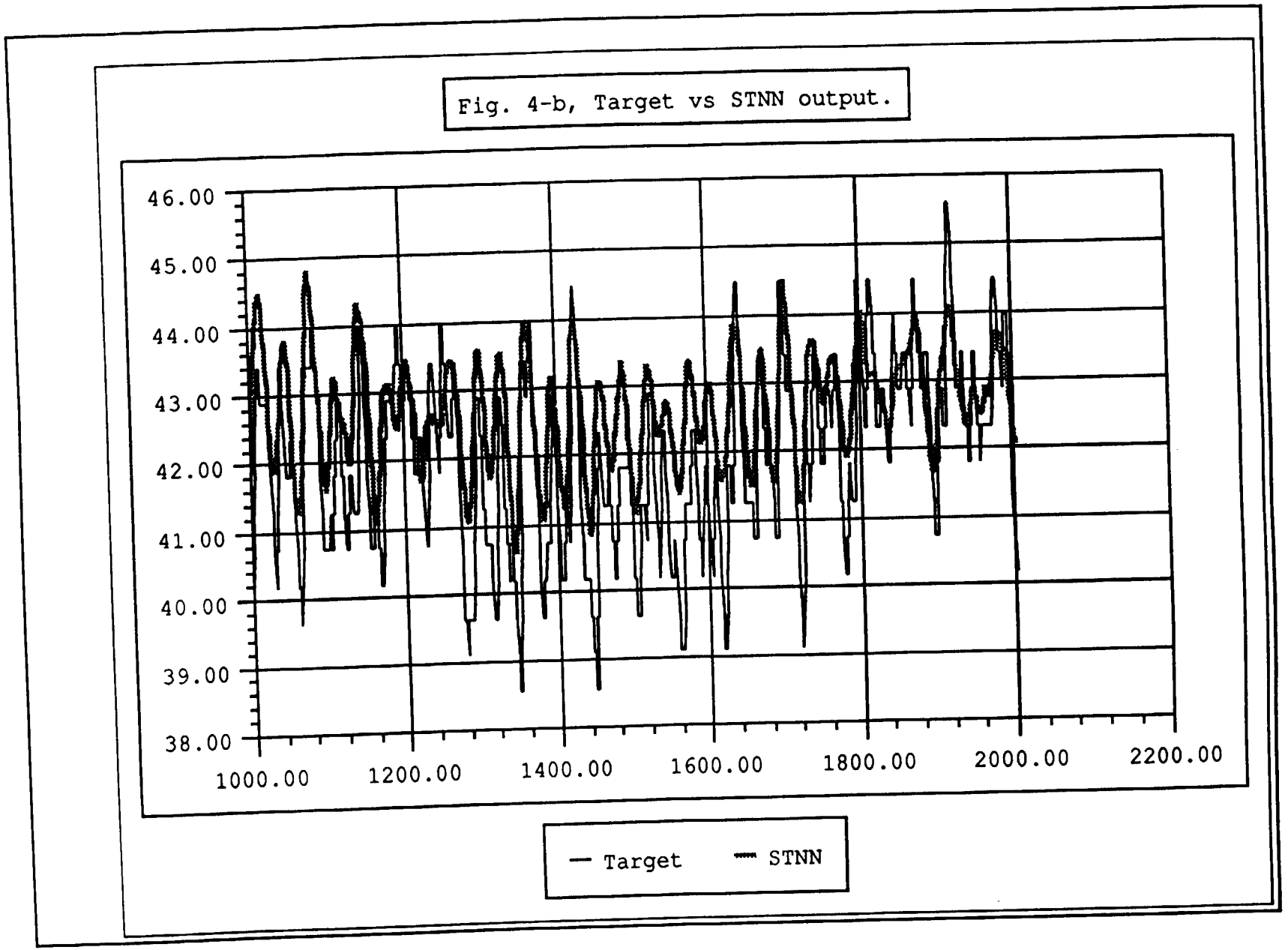


Fig. 4-b, Target vs STNN output.
46.00
45.00
44.00
43.00
42.00
41.00
40.00
39.00
38.00
1000.00
1200.00
1400.00
1600.00
1800.00
2000.00
2200.00
Target
STNN

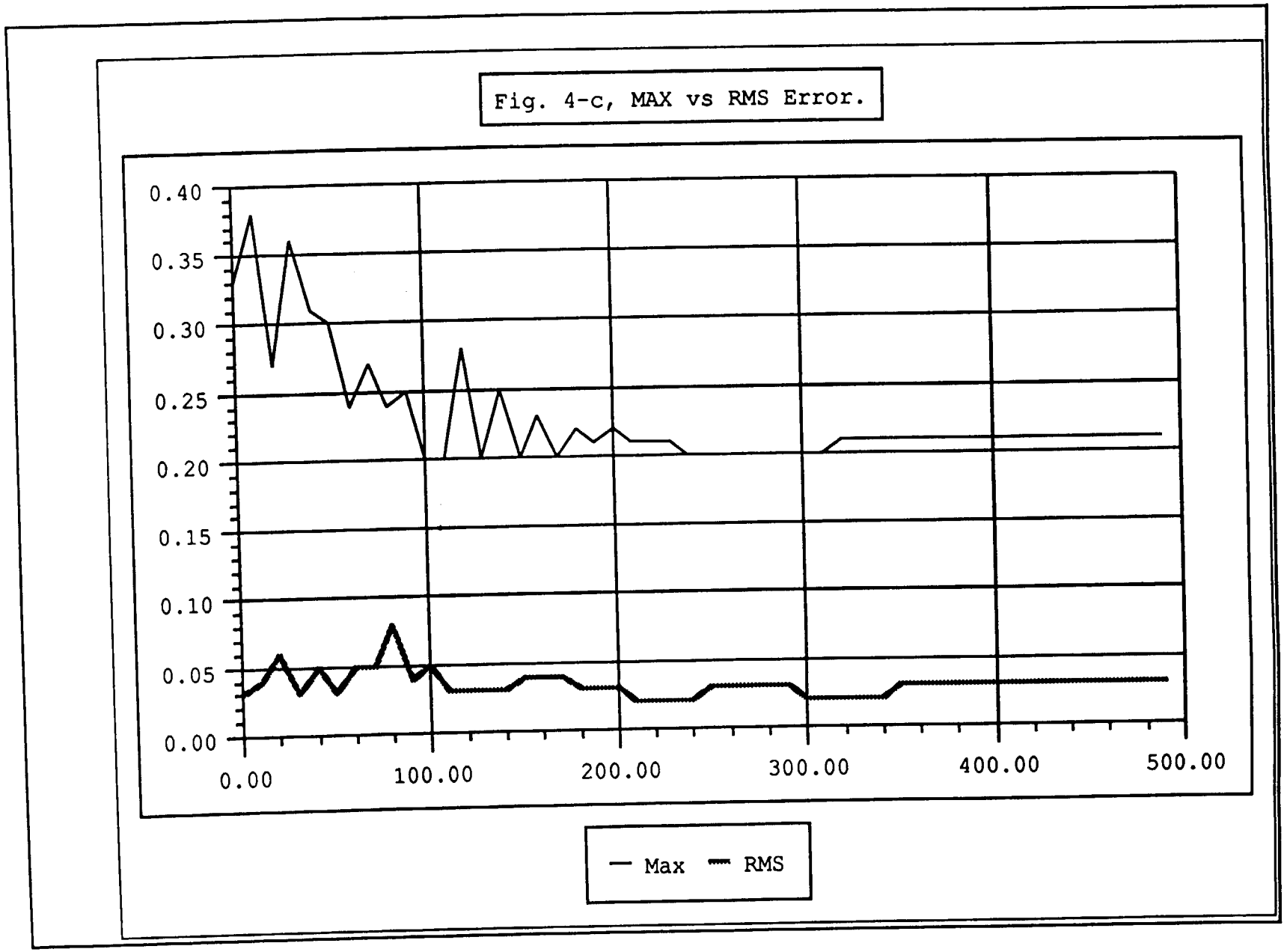


Fig. 4-c, MAX vs RMS Error.
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0.00
100.00
200.00
300.00
400.00
500.00
Max
RMS

test data. The lowest errors were reached after 240 cycles when the maximum error reached 0.2 and the RMS error reached 0.02. Fig. 4-d shows the STNN prediction of skiprope amplitude compared to actual amplitude. Again, performance seems to be within 2-to-3 meters of actual skiprope amplitude. The previous two test cases indicate that the STNN identifies circular skiprope motion within the required 5.0 meters or 10% of actual amplitude, as specified in the Skiprope Observer requirements.

Next, we train and test the STNN on skiprope data corresponding to the motion shown in Fig. 2-b. This motion results from current flowing through the tether during the On-Station-1 portion of the mission. In this test case, we use roll, pitch, and yaw rates, roll, pitch, and yaw angles, sensed length, and sensed tension as inputs, and skiprope amplitude as output. Fig. 4-e shows the MAX and RMS errors reached during training and testing on all 3501 I/O pairs. After 360 cycles, the STNN reached a MAX error of 0.17, and an RMS error of 0.04. Fig. 4-f shows that the STNN seems to have learned the training data after 360 cycles of training.

In our next experiment, we split the data into a training set and a test set by training on the first and last 200 I/O pairs and testing on the middle 2000. Fig. 4-g shows that the errors decrease for only about 50 cycles, and then begin to increase. Fig. 4-h shows that performance after 100 cycles of training is not as good as what was achieved above on circular skiprope data.

In our next experiment, we train and test on data corresponding to the skiprope motion depicted in figure 2-c. This very complex motion results from combinations of current flow and satellite spin throughout satellite deployment and retrieval. In this experiment, we train and test on all I/O pairs (3,502). Fig. 4-i shows that after 40 cycles, the network reached a MAX error of 0.29, and RMS error of 0.05. Fig. 4-j reveals that the network identifies the skiprope amplitude to within 2 meters. Again, for clarity, we only show a portion of the mapping of the entire data set. A plot of the entire data set reveals that the network can be off by as much as 6 meters in some areas.

4.2 Identifying Phase

In this section we examine test cases in which the STNN has been asked to identify skiprope phase in addition to amplitude. As in the previous section, we start with a circular skiprope motion and progress to more difficult situations. In our first experiment, we use roll and pitch rates, roll and pitch angles, sensed length, and sensed tension as input, and produce amplitude and phase on the outputs. In addition to the 6 inputs, and two outputs, the network consists of thirty hidden units, and forty filters between input and hidden, and hidden and output layers. Fig. 5-a shows the MAX and RMS errors achieved as the network trained on the first and last 200 I/O pairs, and tested on the middle 3,600 I/O pairs from the full set of 4,001 I/O pairs. Fig. 5-b shows a portion of the network's estimation of skiprope amplitude. The performance of the network is generally within 6 meters over the entire test data set. Fig. 5-c shows that the network identifies skiprope phase to within 50 degrees, which is not within the required 10 degrees.

Subsequent efforts to identify skiprope phase also fall short of the requirements. Fig. 5-d shows the training errors resulting from an attempt to train on data corresponding to a skiprope motion resulting from satellite spin and current flow through the tether. In this test case, the network trained and tested on a complete set of 3,501 I/O pairs. Although Fig. 5-e shows that the network identifies amplitude to within 4 meters, the network may incorrectly identify amplitude by as much as 8 meters. Fig. 5-f shows that the network performs poorly in identifying skiprope phase.

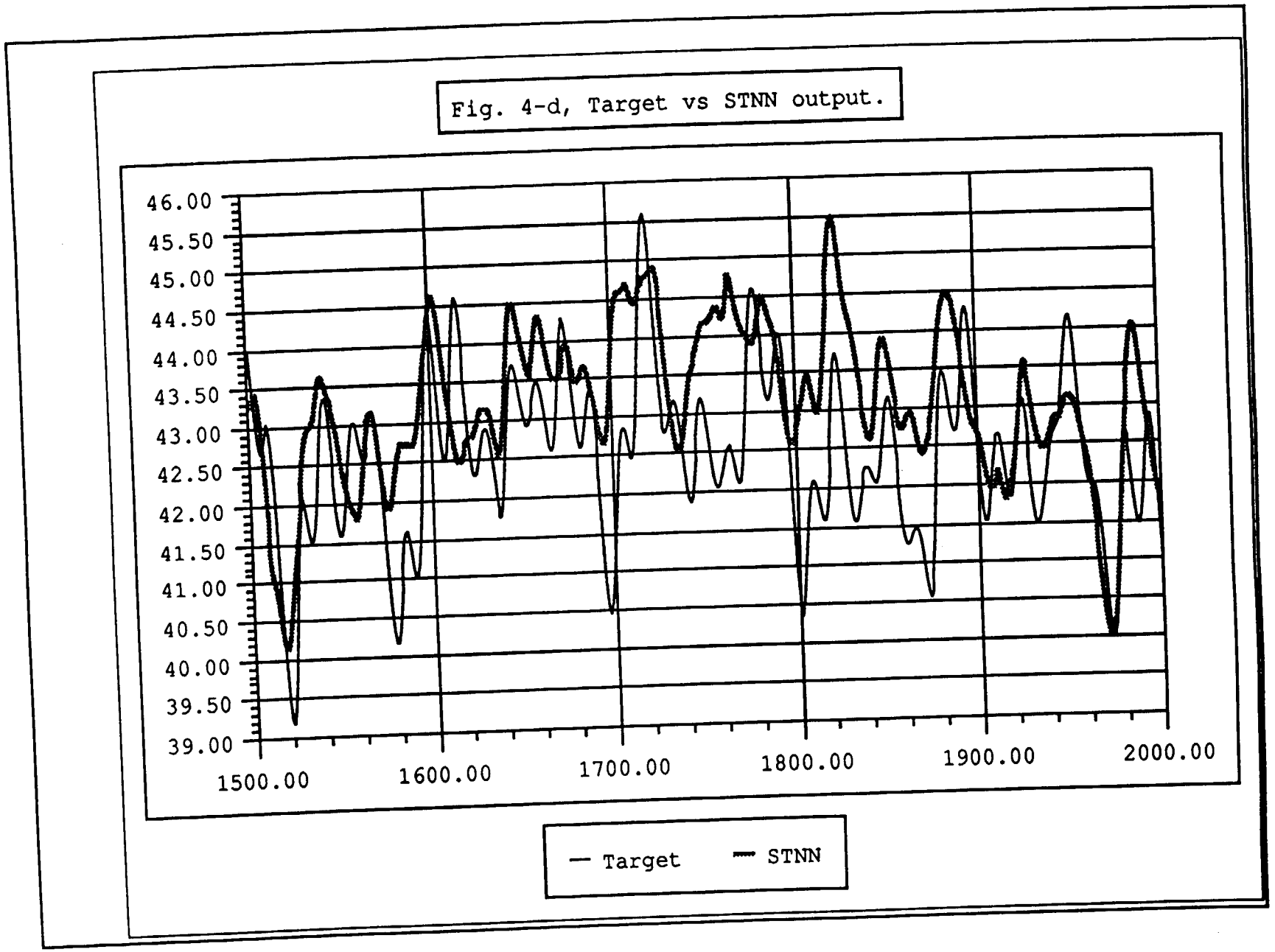


Fig. 4-d, Target vs STNN output.
46.00
45.50
45.00
44.50
44.00
43.50
43.00
42.50
42.00
41.50
41.00
40.50
40.00
39.50
39.00
1500.00
1600.00
1700.00
1800.00
1900.00
2000.00
Target
STNN

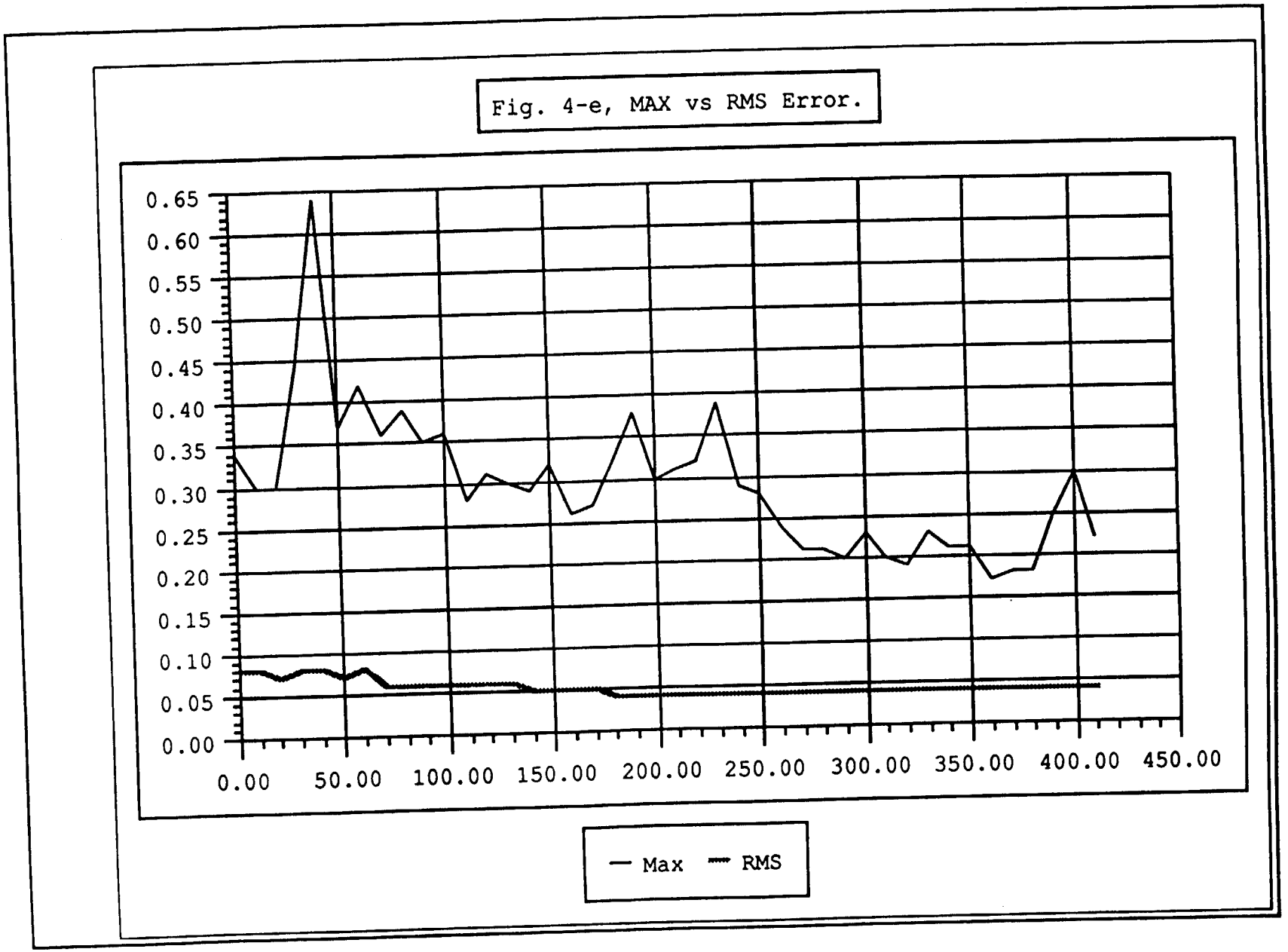


Fig. 4-e, MAX vs RMS Error.
0.65
0.60
0.55
0.50
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0.00
50.00
100.00
150.00
200.00
250.00
300.00
350.00
400.00
450.00
Max
RMS

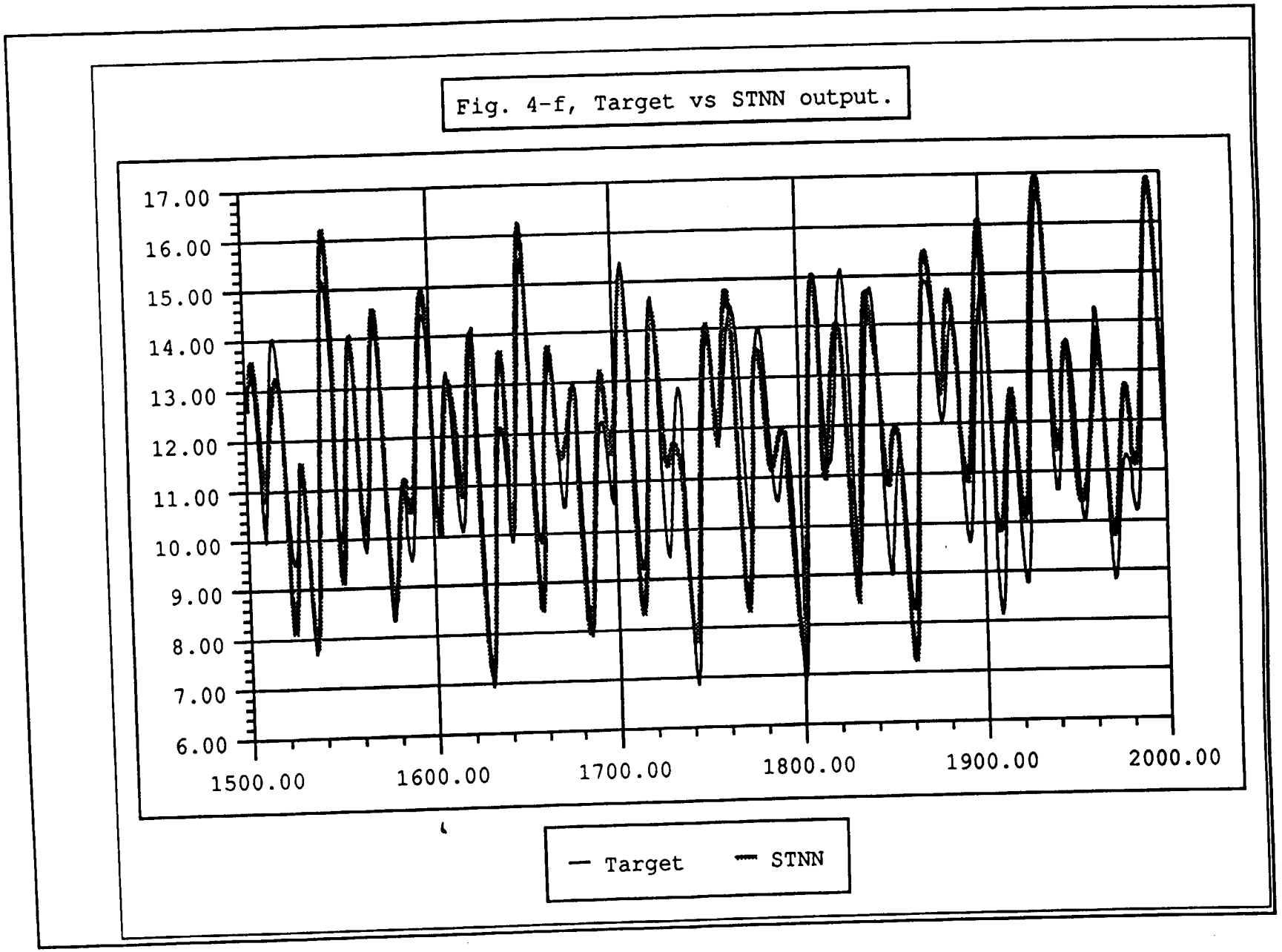


Fig. 4-f, Target vs STNN output.
17.00
16.00
15.00
14.00
13.00
12.00
11.00
10.00
9.00
8.00
7.00
6.00
1500.00
1600.00
1700.00
1800.00
1900.00
2000.00
Target
STNN

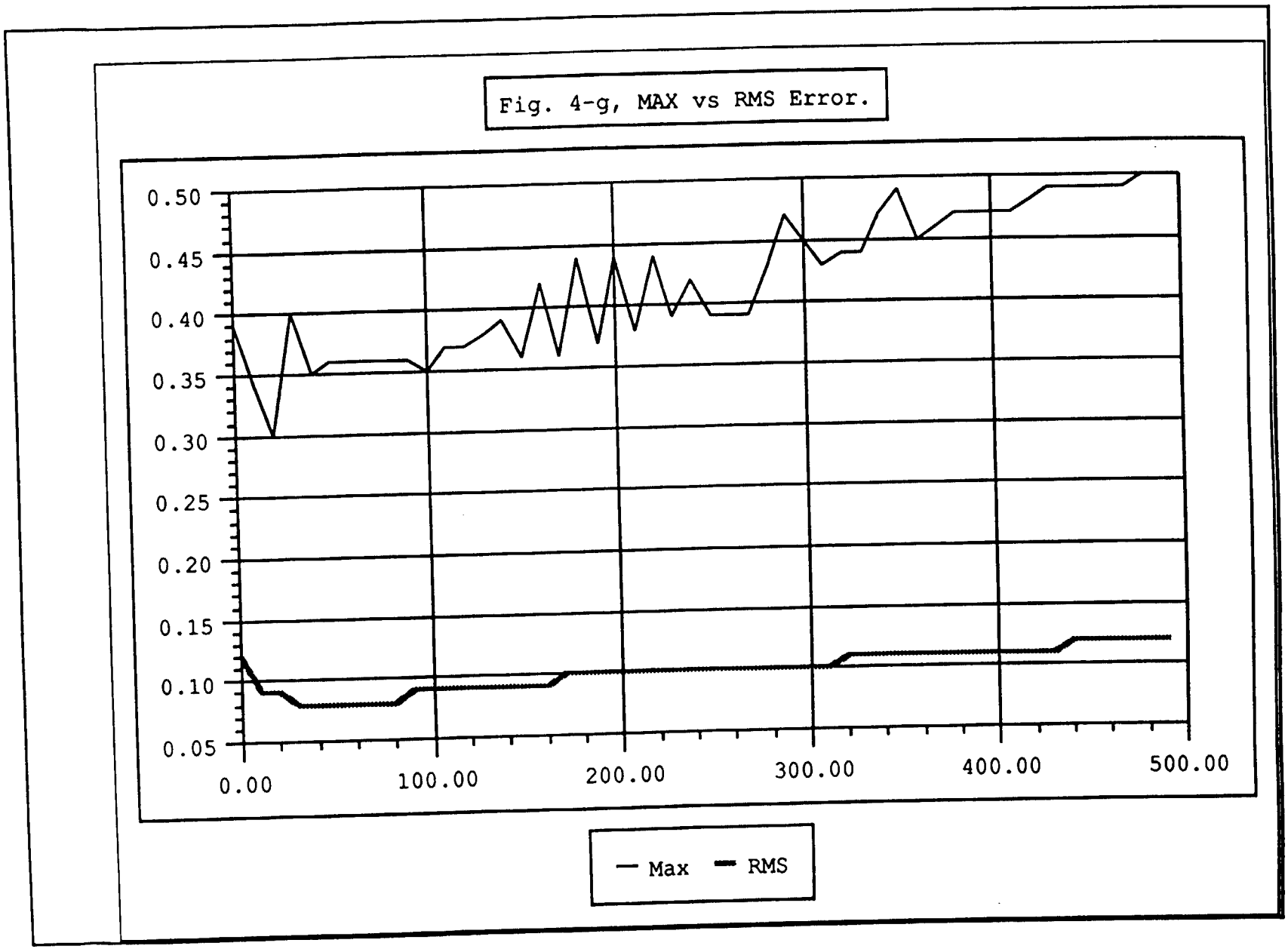


Fig. 4-g, MAX vs RMS Error.
0.50
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
100.00
200.00
300.00
400.00
500.00
Max
RMS

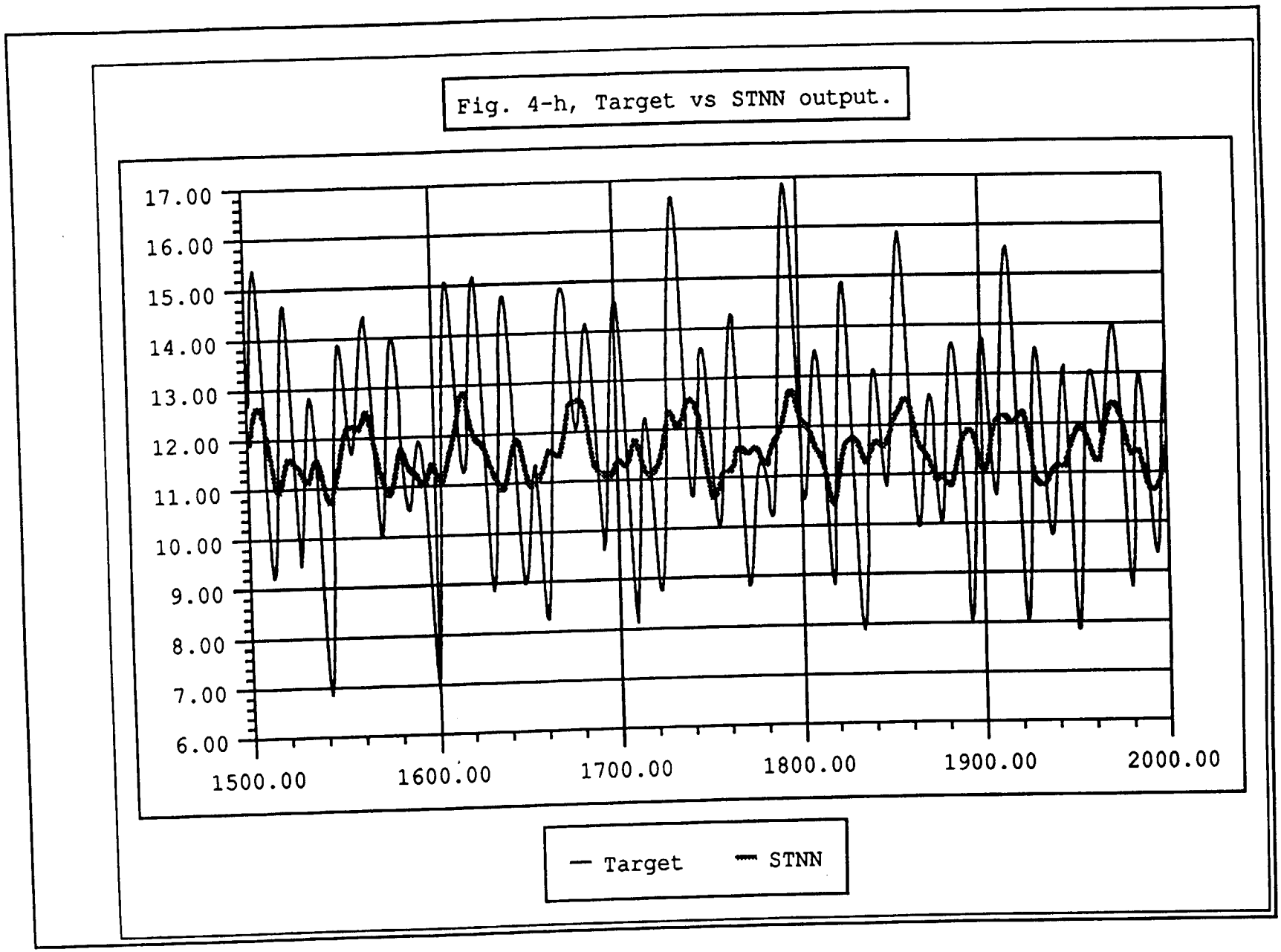


Fig. 4-h, Target vs STNN output.
17.00
16.00
15.00
14.00
13.00
12.00
11.00
10.00
9.00
8.00
7.00
6.00
1500.00
1600.00
1700.00
1800.00
1900.00
2000.00
Target
STNN

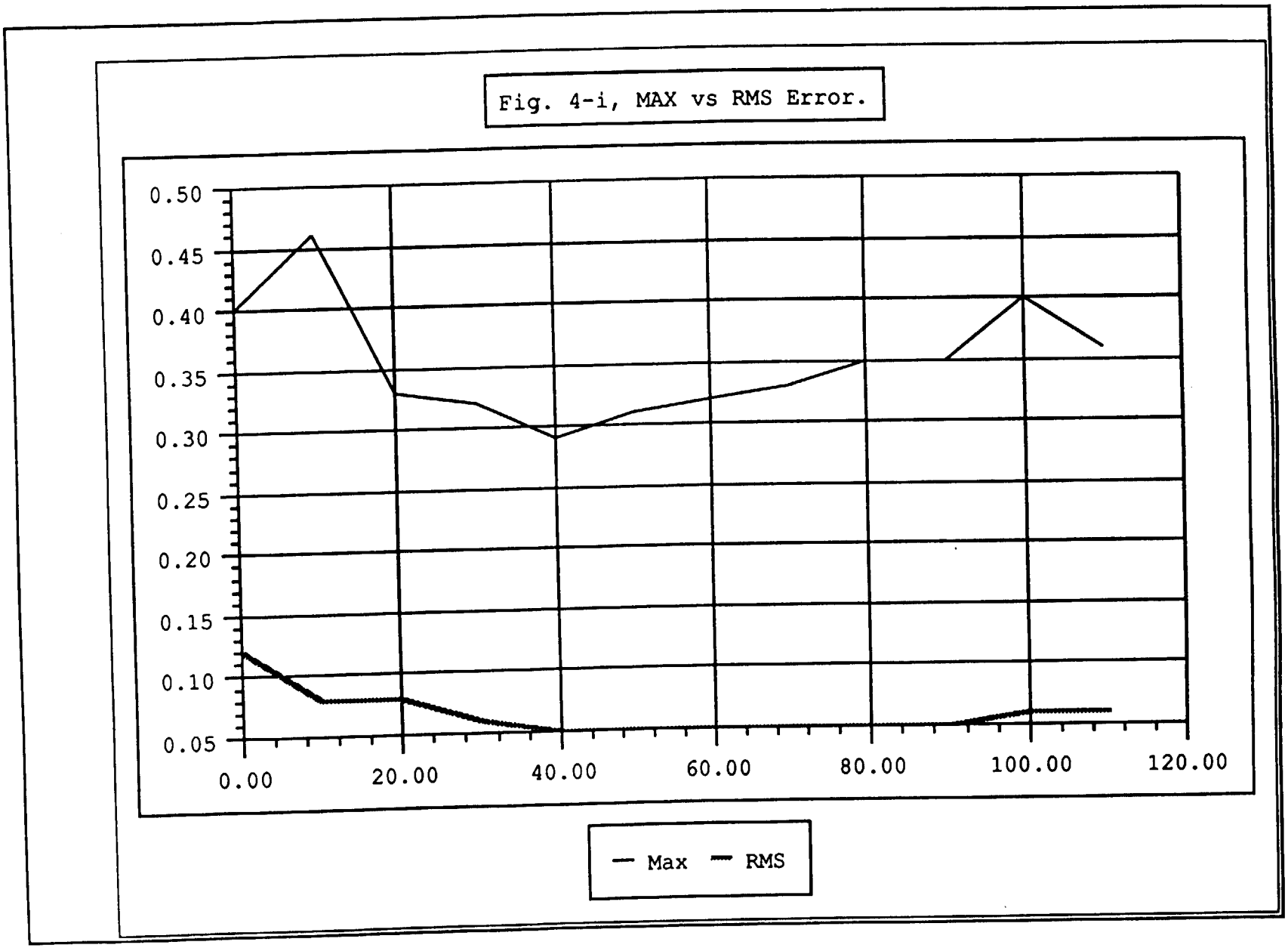


Fig. 4-i, MAX vs RMS Error.
0.50
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
20.00
40.00
60.00
80.00
100.00
120.00
Max
RMS

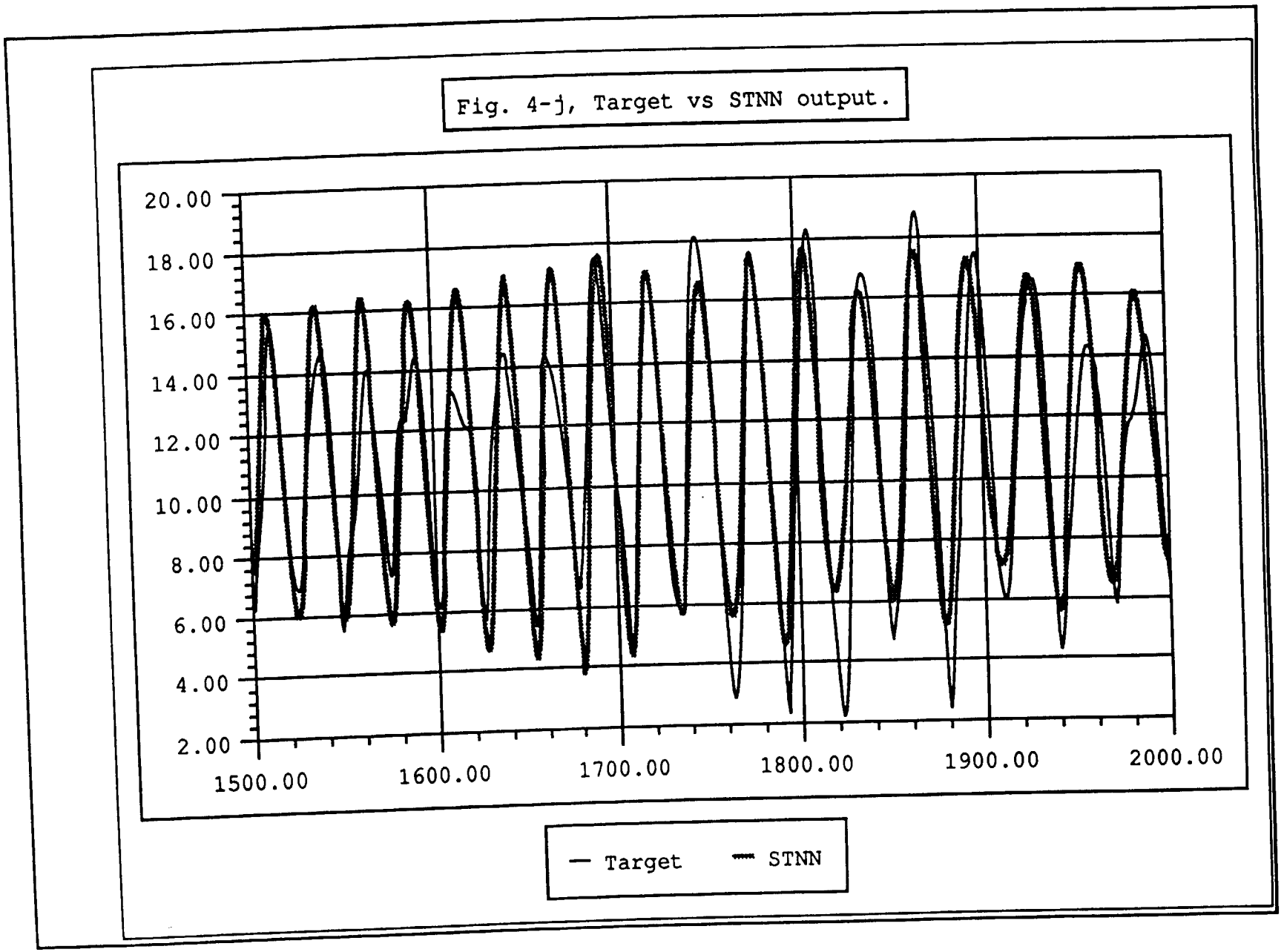


Fig. 4-j, Target vs STNN output.
20.00
18.00
16.00
14.00
12.00
10.00
8.00
6.00
4.00
2.00
1500.00
1600.00
1700.00
1800.00
1900.00
2000.00
Target
STNN

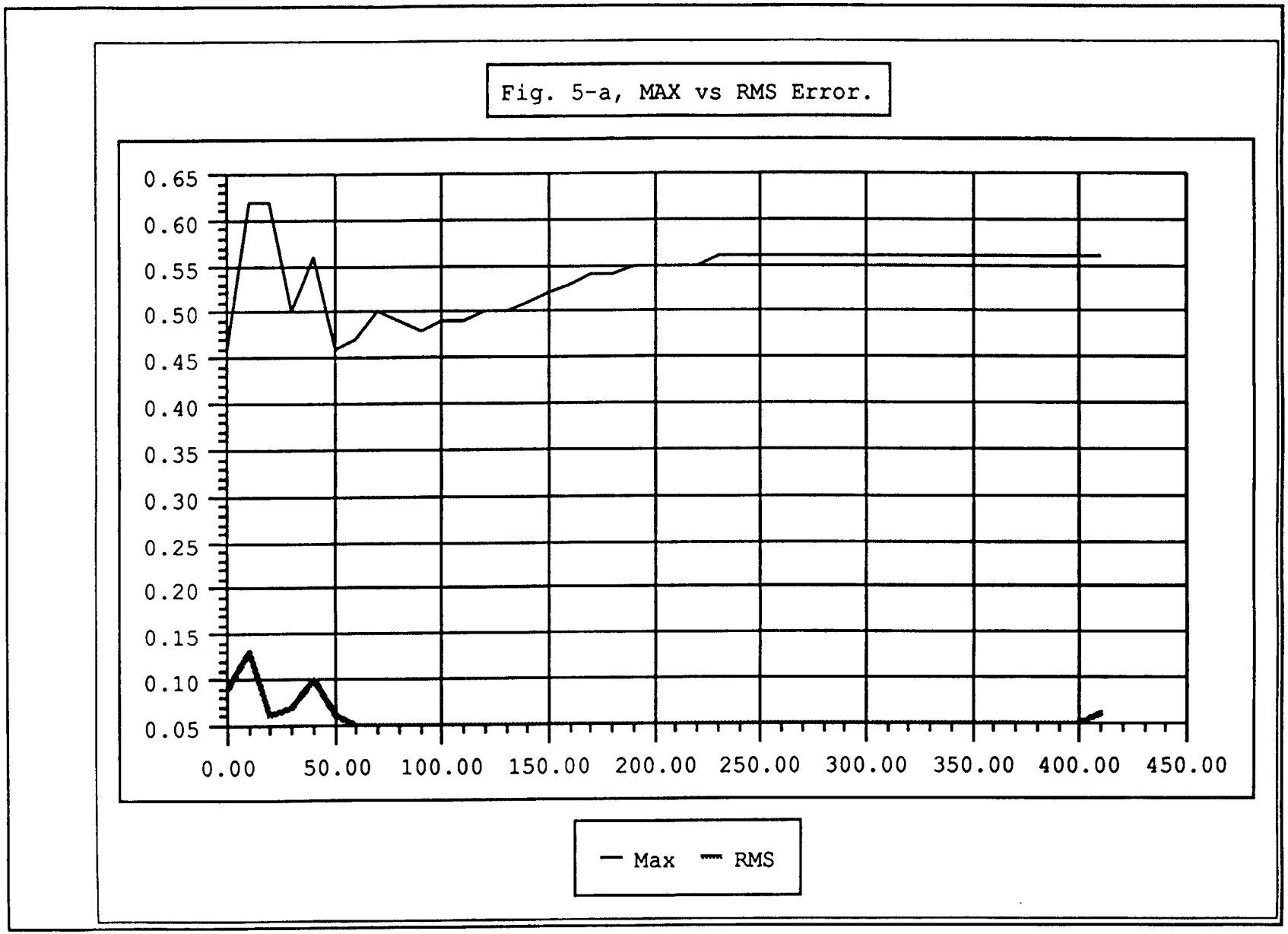


Fig. 5-a, MAX vs RMS Error.
0.65
0.60
0.55
0.50
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
50.00
100.00
150.00
200.00
250.00
300.00
350.00
400.00
450.00
Max
RMS

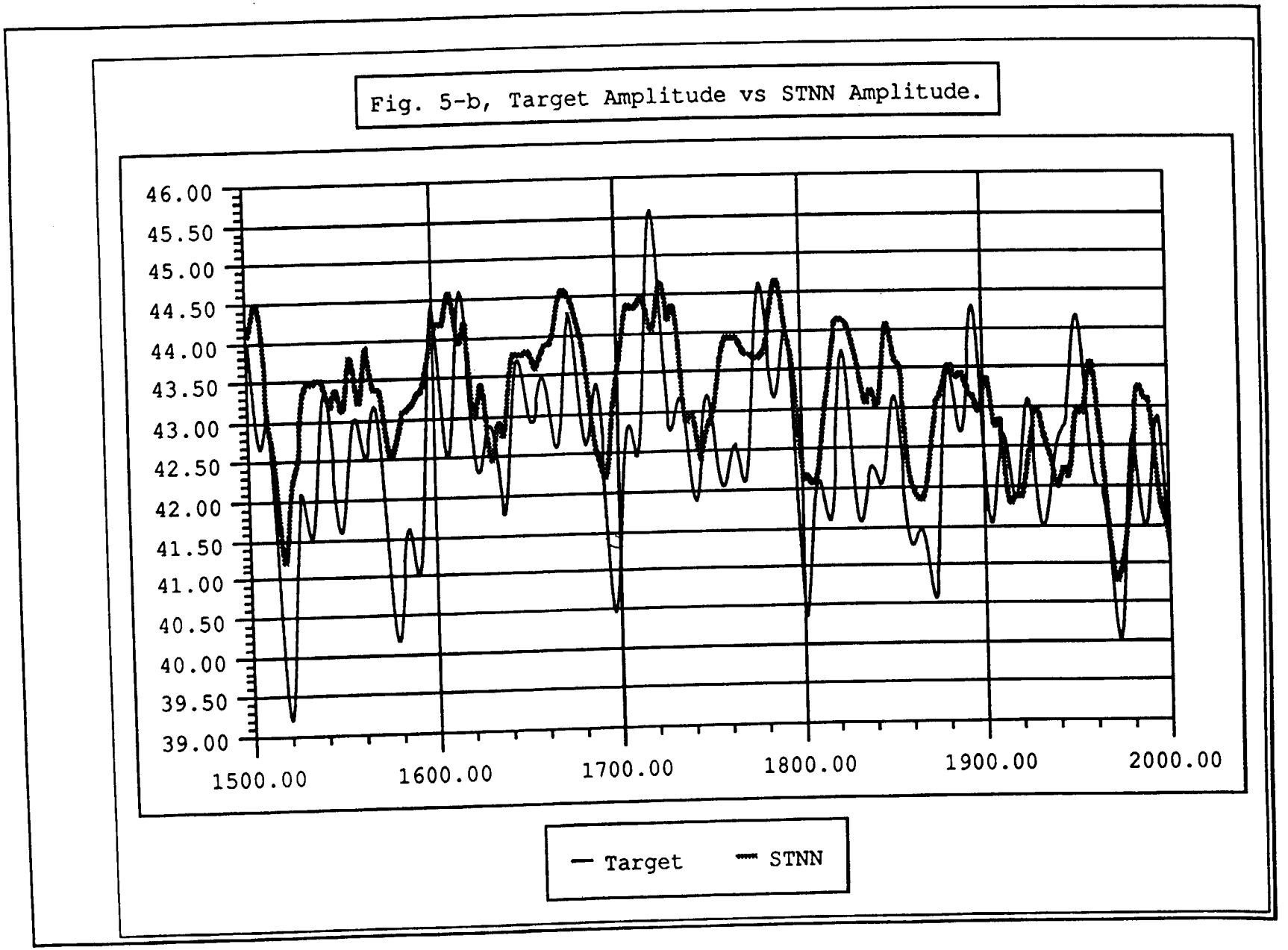


Fig. 5-b, Target Amplitude vs STNN Amplitude.
46.00
45.50
45.00
44.50
44.00
43.50
43.00
42.50
42.00
41.50
41.00
40.50
40.00
39.50
39.00
1500.00
1600.00
1700.00
1800.00
1900.00
2000.00
Target
STNN

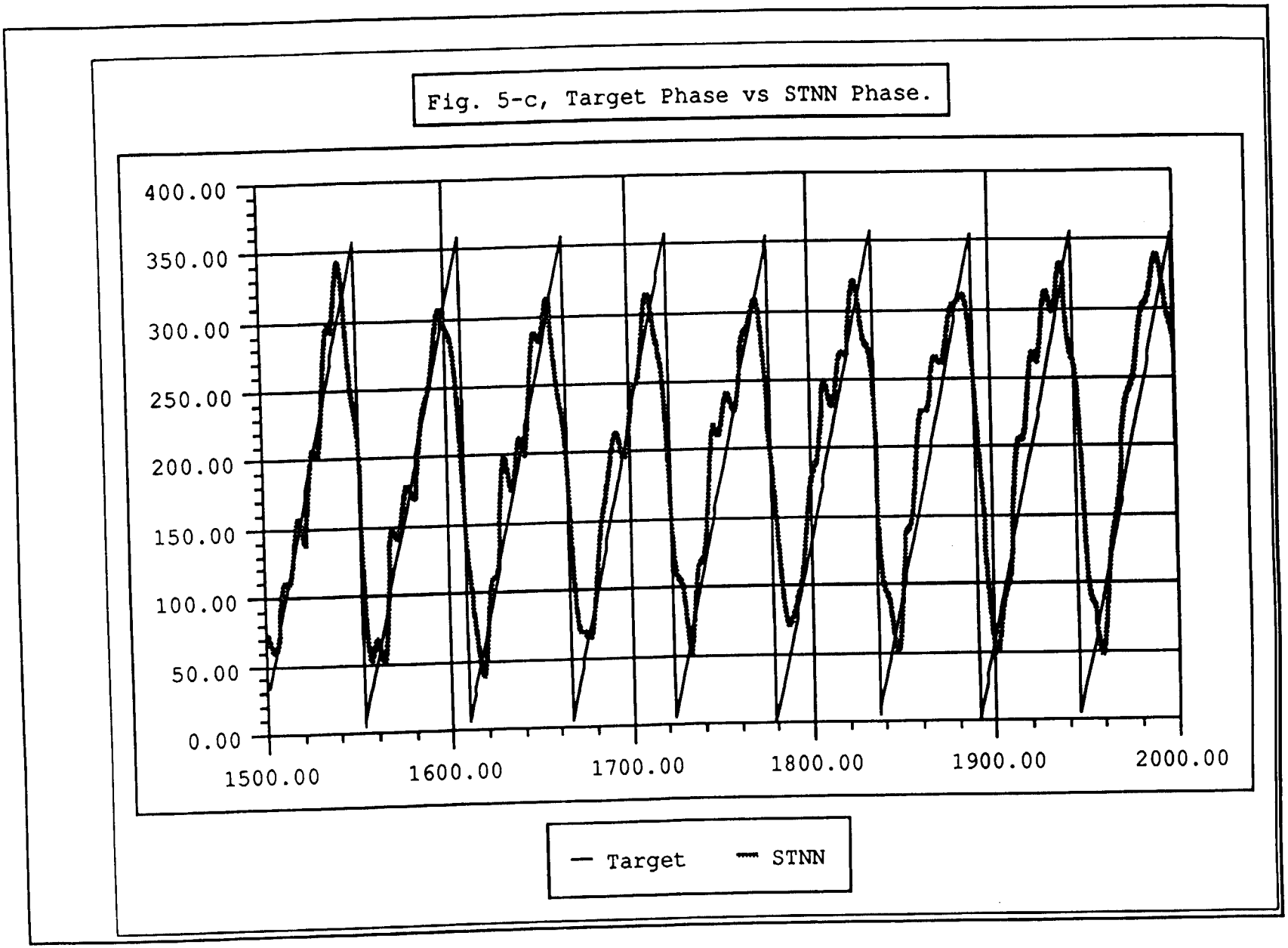


Fig. 5-c, Target Phase vs STNN Phase.
400.00
350.00
300.00
250.00
200.00
150.00
100.00
50.00
0.00
1500.00
1600.00
1700.00
1800.00
1900.00
2000.00
Target
STNN

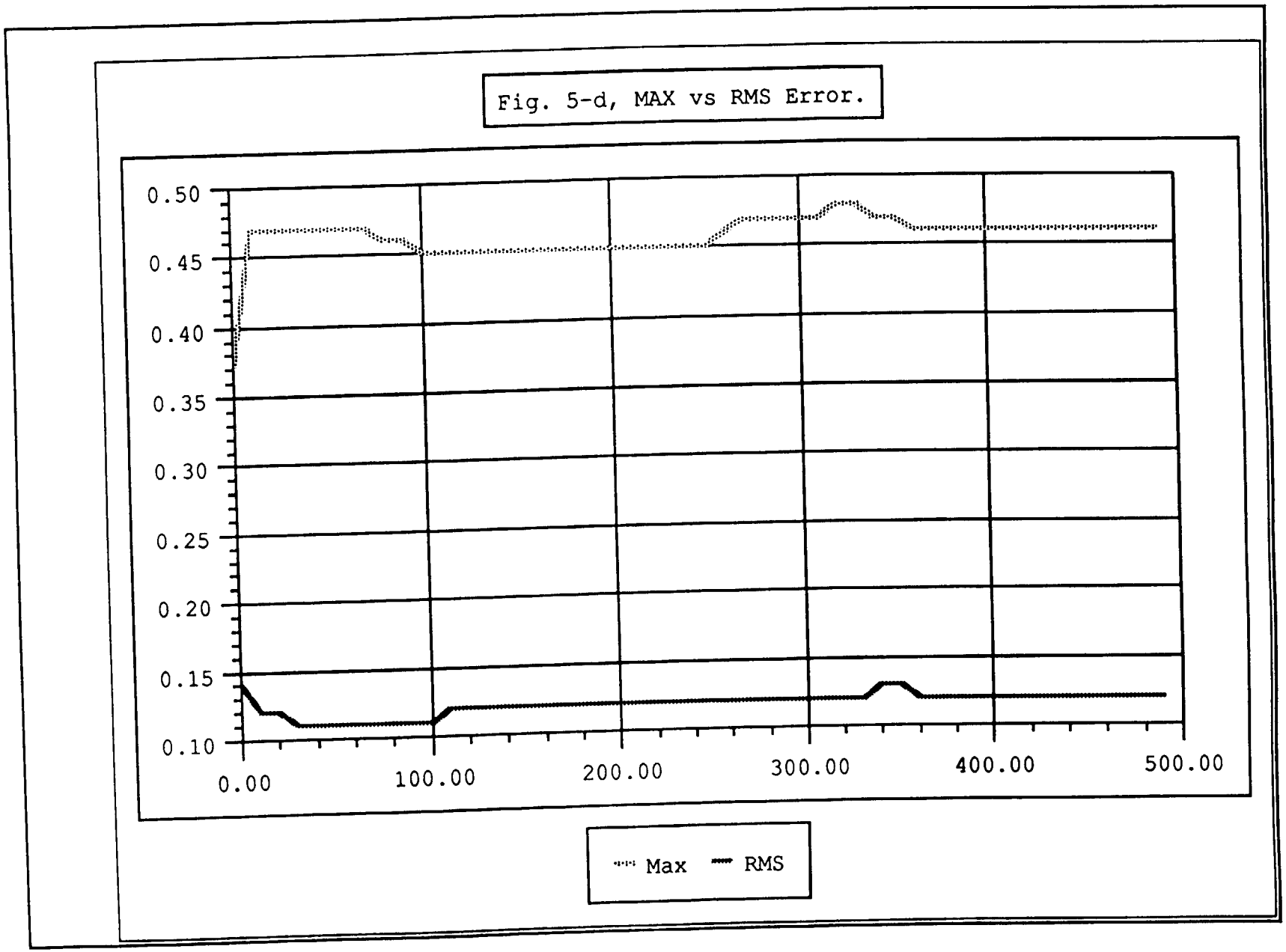


Fig. 5-d, MAX vs RMS Error.
0.50
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.00
100.00
200.00
300.00
400.00
500.00
Max
RMS

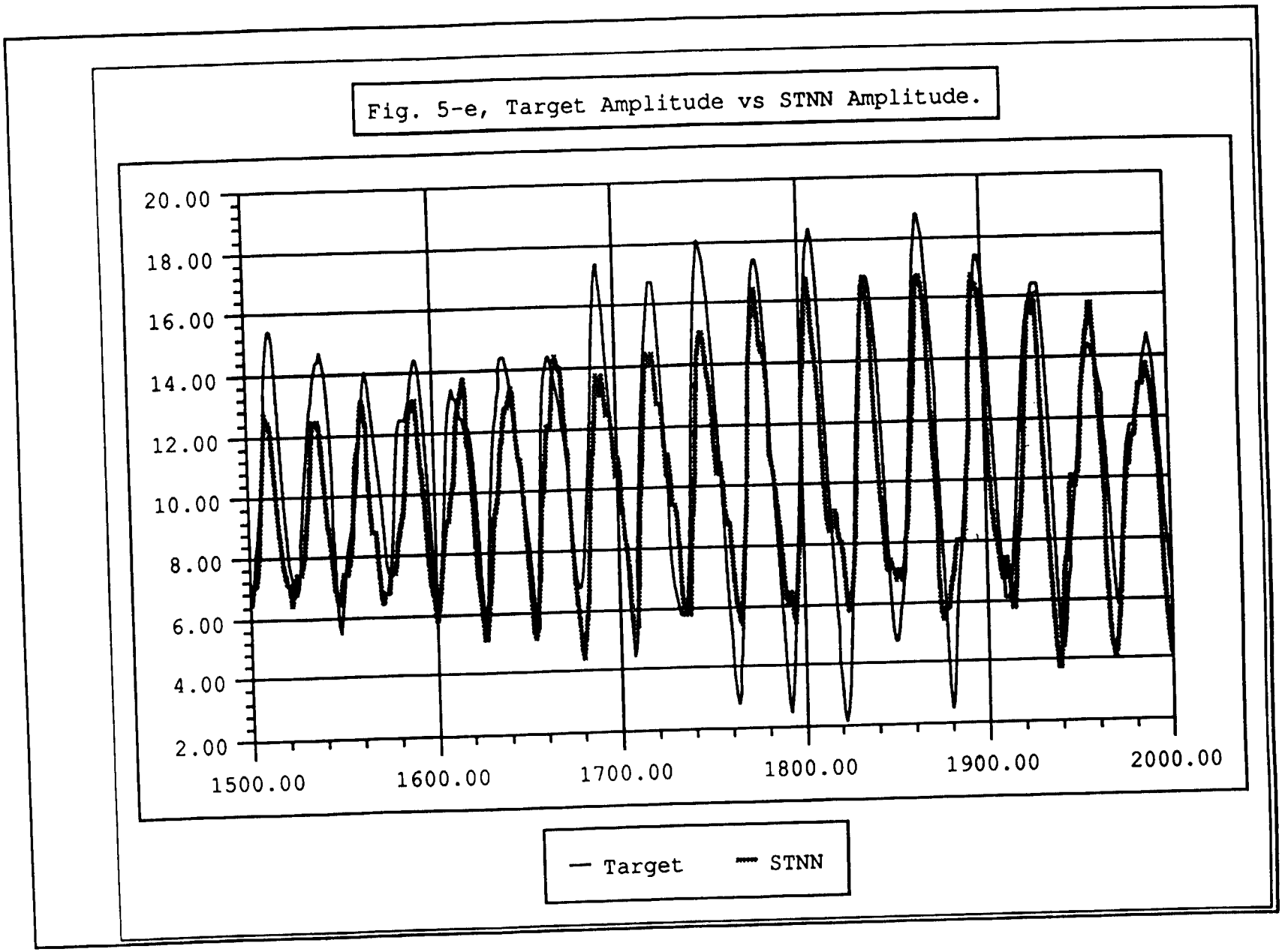


Fig. 5-e, Target Amplitude vs STNN Amplitude.
20.00
18.00
16.00
14.00
12.00
10.00
8.00
6.00
4.00
2.00
1500.00
1600.00
1700.00
1800.00
1900.00
2000.00
Target
STNN

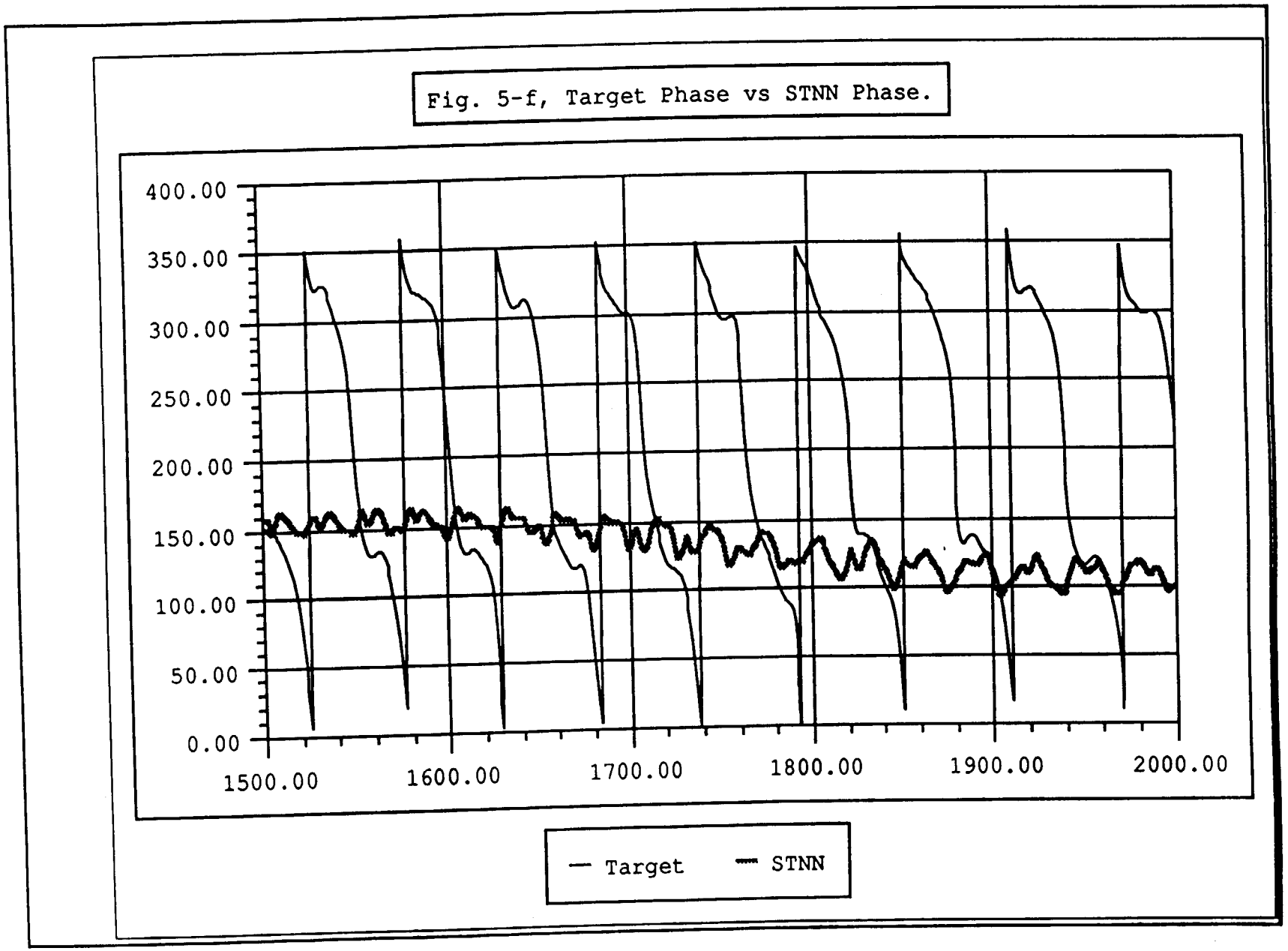


Fig. 5-f, Target Phase vs STNN Phase.
400.00
350.00
300.00
250.00
200.00
150.00
100.00
50.00
0.00
1500.00
1600.00
1700.00
1800.00
1900.00
2000.00
Target
STNN

4.3 Identifying X, Y Components of Skiprope Amplitude

The biggest challenge to network training so far has been to learn the phase mapping. Several different network configurations have yielded good results in predicting skiprope amplitude, but we have not been as lucky with skiprope phase. Since the ultimate goal is to provide the crew with a reasonable estimate of skiprope amplitude and phase to support the yaw maneuver, the skiprope observer should learn not only to identify but also to predict amplitude and phase based on the available inputs. For predicting the skiprope motion, one can use the past estimates of the amplitude and phase and thus the network will have a feedback of its output as shown in Fig. 6-a. In other words, the characteristics of the skiprope motion can be identified based on several parameters that include the past x and y coordinates of the mid-point of the tether during skiprope motion.

The networks in the following test cases use satellite rates (roll, pitch, and yaw), sensed tension, and current x and y coordinates of the mid-point of the tether as inputs, and produce the next x and y position, x(t + 1), y(t + 1). Fig. 6-b shows the MAX and RMS errors achieved while training and testing on all 3,500 I/O pairs. As Fig. 6-b shows, the network reaches a low RMS error of 0.01, and a low MAX error of 0.05 within 500 training cycles. Figs. 6-c, and 6-d show that the network produces an accurate estimation of x and y components of the skiprope motion. Next, we divide the data into a training set and a test set and test for network generalization. Fig. 6-e shows the MAX and RMS test errors achieved after training on the first and last 400 I/O pairs, and testing on the middle 2,700 I/O pairs. Figs. 6-f and 6-g show that the network performed well on the test set. In reality, it may be impractical to use current x and y on the inputs to the network, so in subsequent test cases, we have used only satellite rates (roll, pitch, and yaw), satellite angles (roll, pitch, and yaw), sensed length, and sensed tension as inputs and trained the network to output x and y.

4.4 Combined Test Cases

So far we have focussed our efforts on training an STNN based skiprope observer to perform based on inputs representing one type of skiprope motion at a time. However, in order to place a neural network based skiprope observer in an operational environment, we must ensure that the network can be trained on data representing many different scenarios and perform adequately on conditions that it may have never seen. In the test cases described above we divided data sets into training sets and test sets to test for generalization. However, these experiments only tested the networks ability to generalize on data that was consistent with the training data. In the following test case, we train on part of the data from a simulation containing current flow and satellite spin, and data from a simulation with partial current flow and no satellite spin. The network is then tested on data that it has not seen from a simulation containing satellite spin and current flow. This method of testing ensures that the test data is consistent with some, but not all of the training data. As Fig. 7-a shows, the network reaches a low test set MAX error of 0.48 and RMS error of 0.12 after 150 cycles. Figs. 7-b and 7-c show that the network performs poorly in identifying skiprope X and Y components in this experiment.

5.0 Advantages and disadvantages of STNN over other methods

The primary skiprope detection system developed for the TSS-1 flight uses a ground-based Kalman filter coupled with a one-bead finite element model of the tethered system. The filter estimates amplitude, phase, and frequency of the skiprope motion based on the downlinked telemetry data. The simulation uses the downlinked satellite rate gyro data to

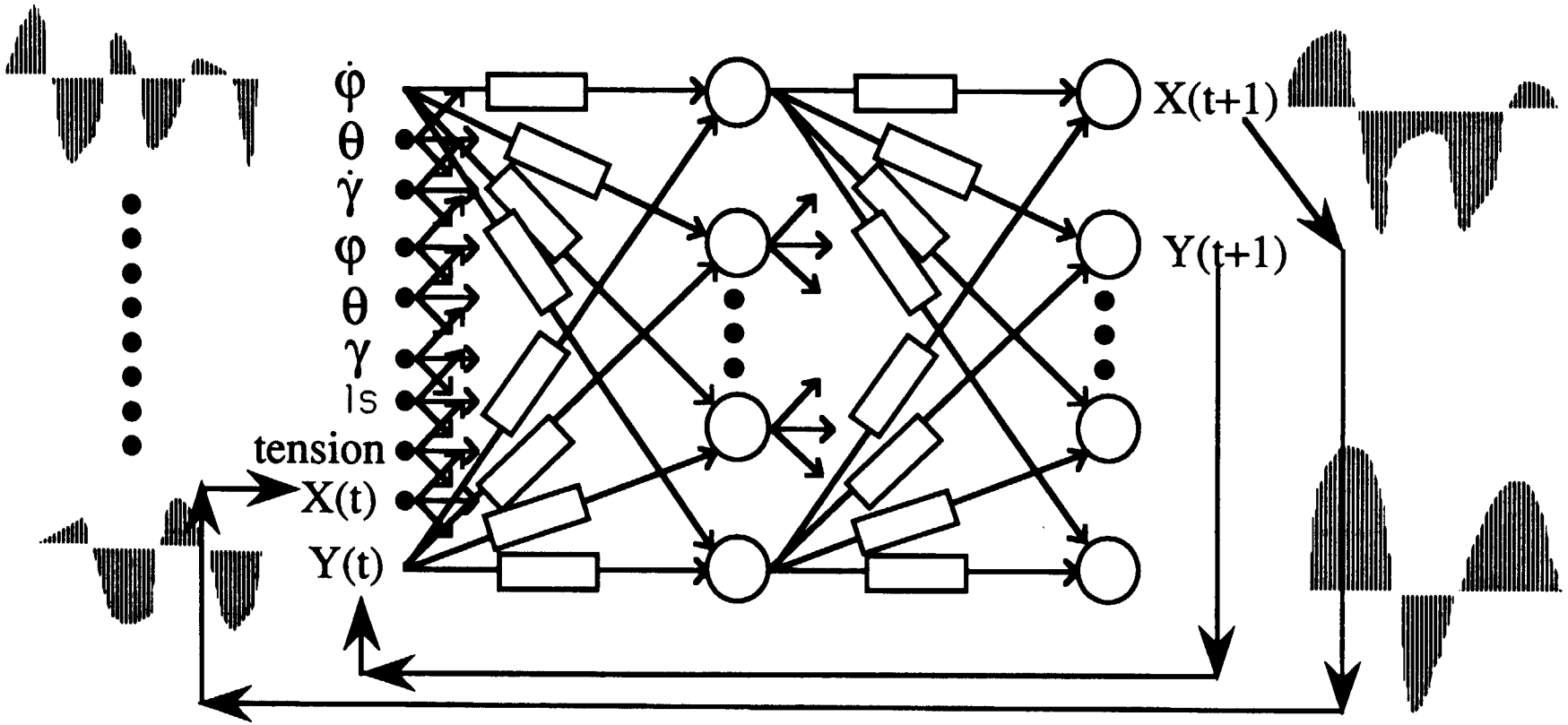


Fig. 6-a STNN Configuration with X(t) and Y(t) as feedback parameters

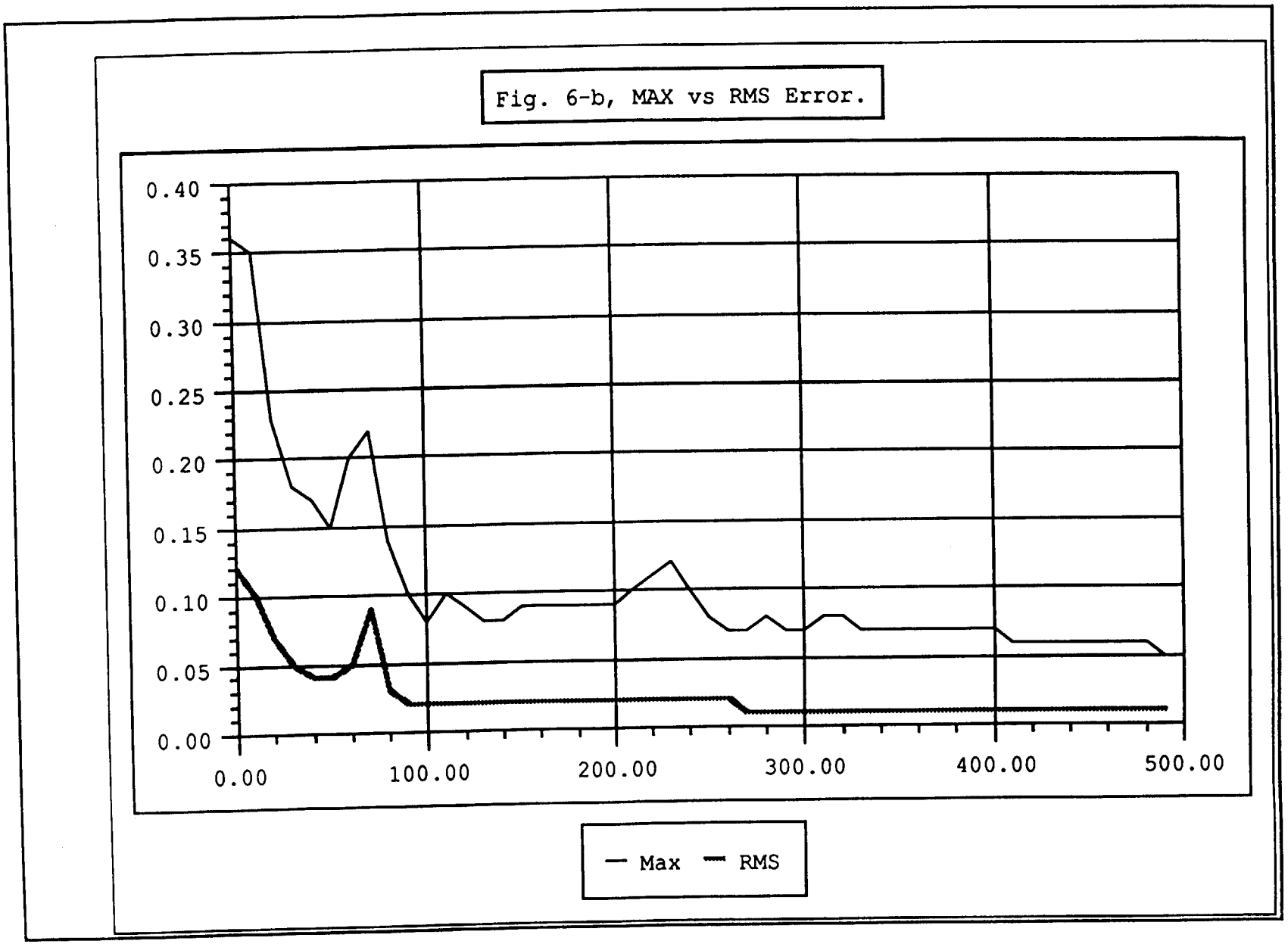


Fig. 6-b, MAX vs RMS Error.
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0.00
100.00
200.00
300.00
400.00
500.00
Max
RMS

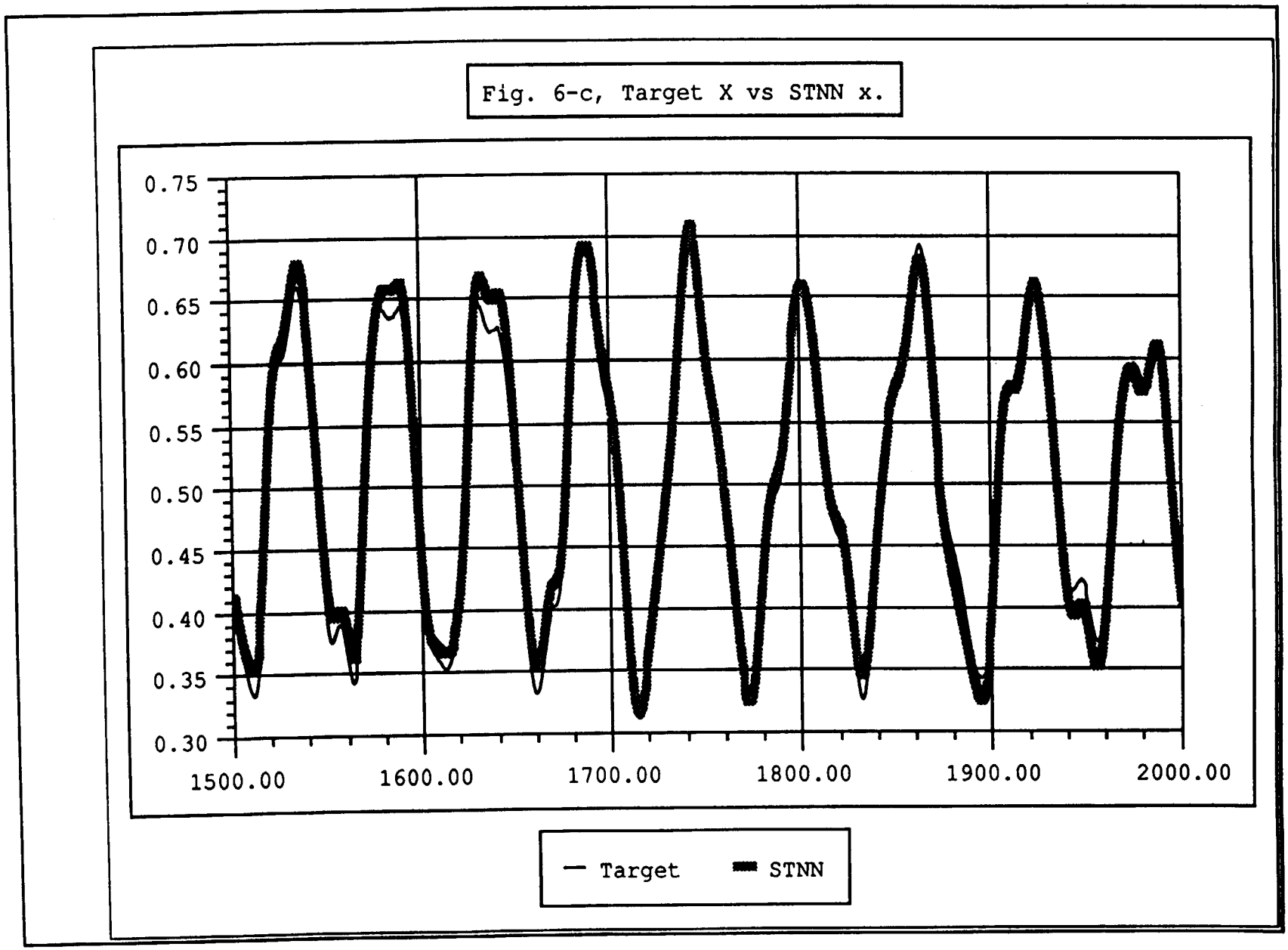


Fig. 6-c, Target X vs STNN x.
0.75
0.70
0.65
0.60
0.55
0.50
0.45
0.40
0.35
0.30
1500.00
1600.00
1700.00
1800.00
1900.00
2000.00
Target
STNN

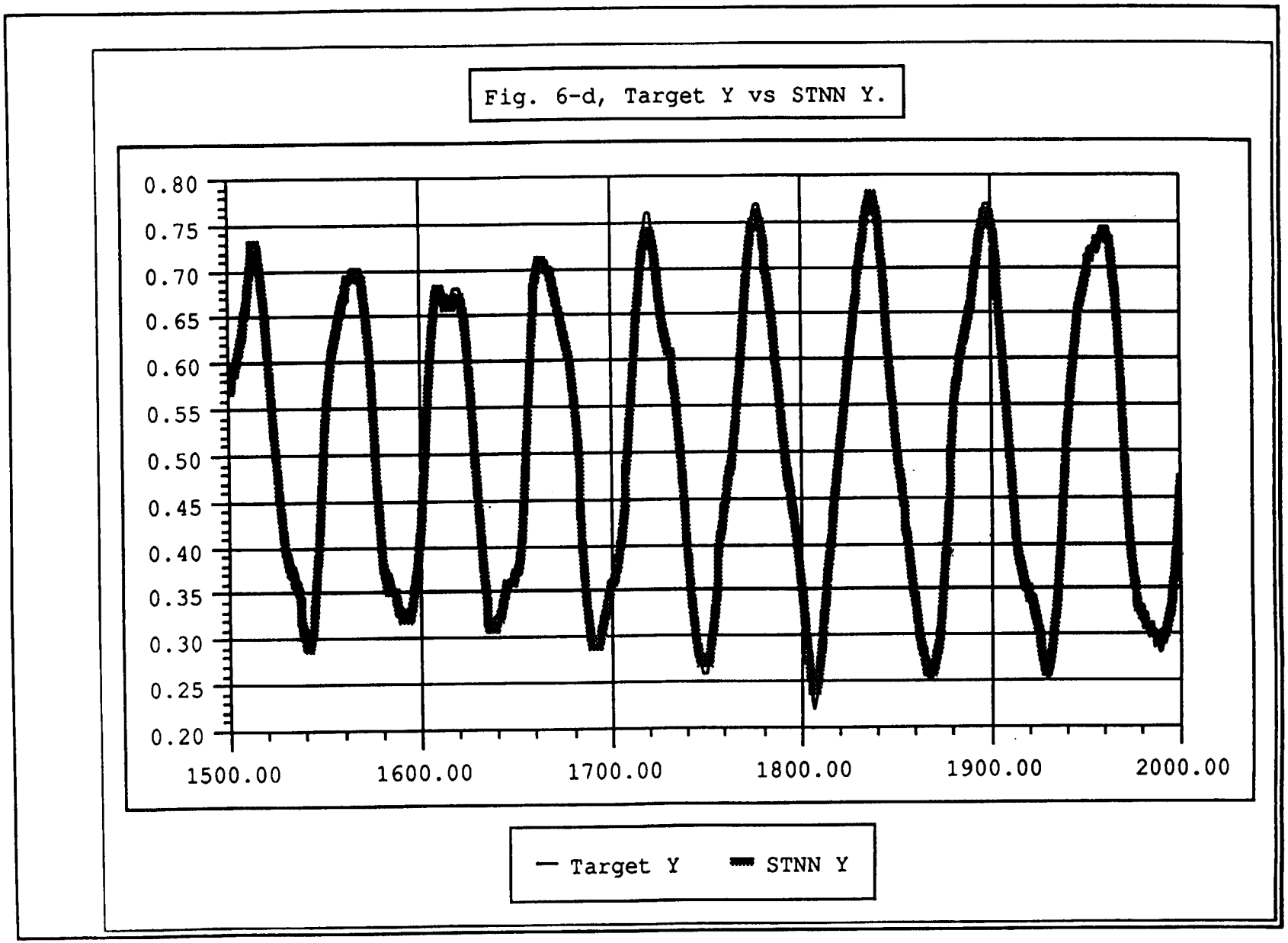


Fig. 6-d, Target Y vs STNN Y.
0.80
0.75
0.70
0.65
0.60
0.55
0.50
0.45
0.40
0.35
0.30
0.25
0.20
1500.00
1600.00
1700.00
1800.00
1900.00
2000.00
Target Y
STNN Y

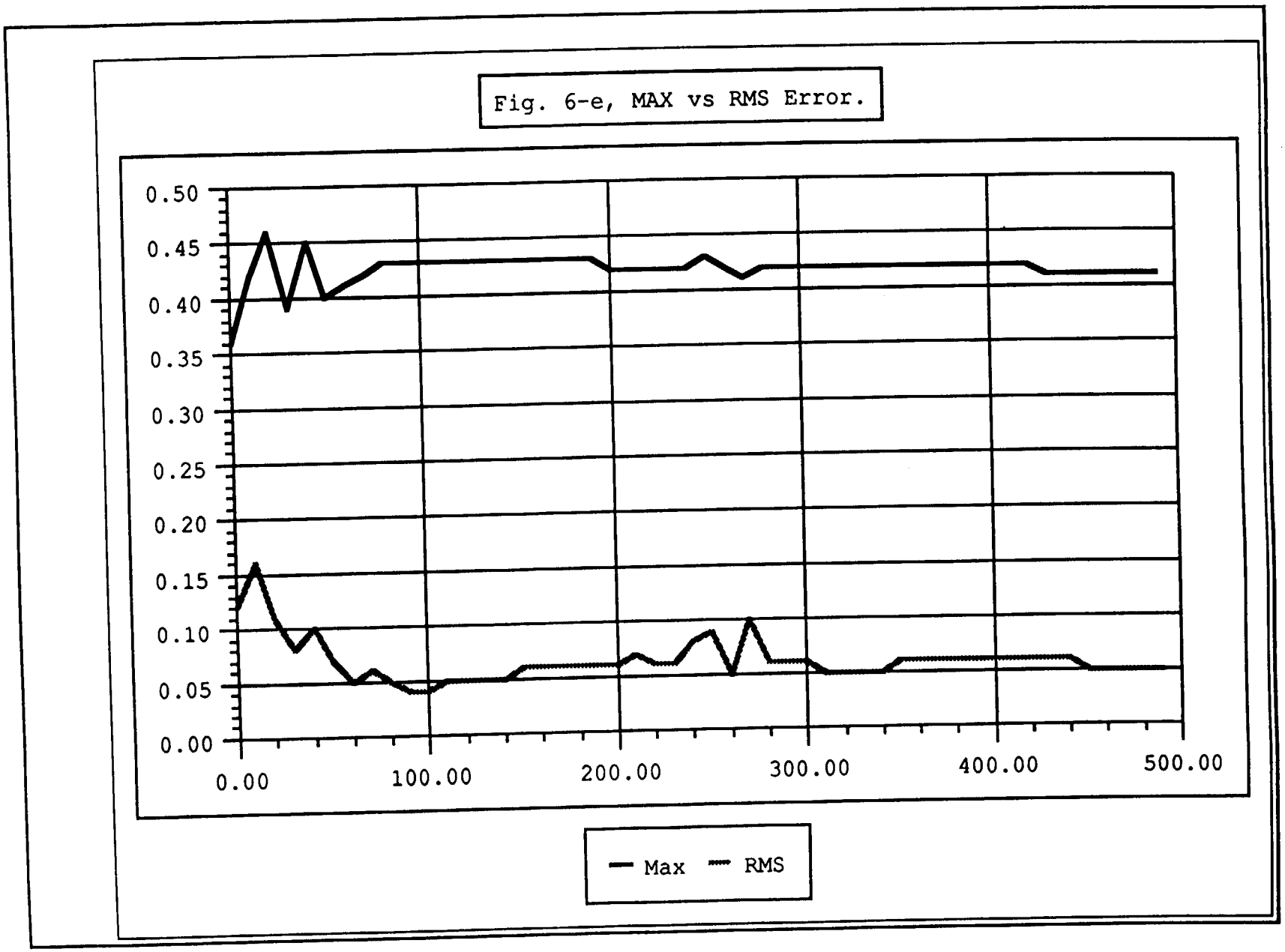


Fig. 6-e, MAX vs RMS Error.
0.50
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0.00
100.00
200.00
300.00
400.00
500.00
Max
RMS

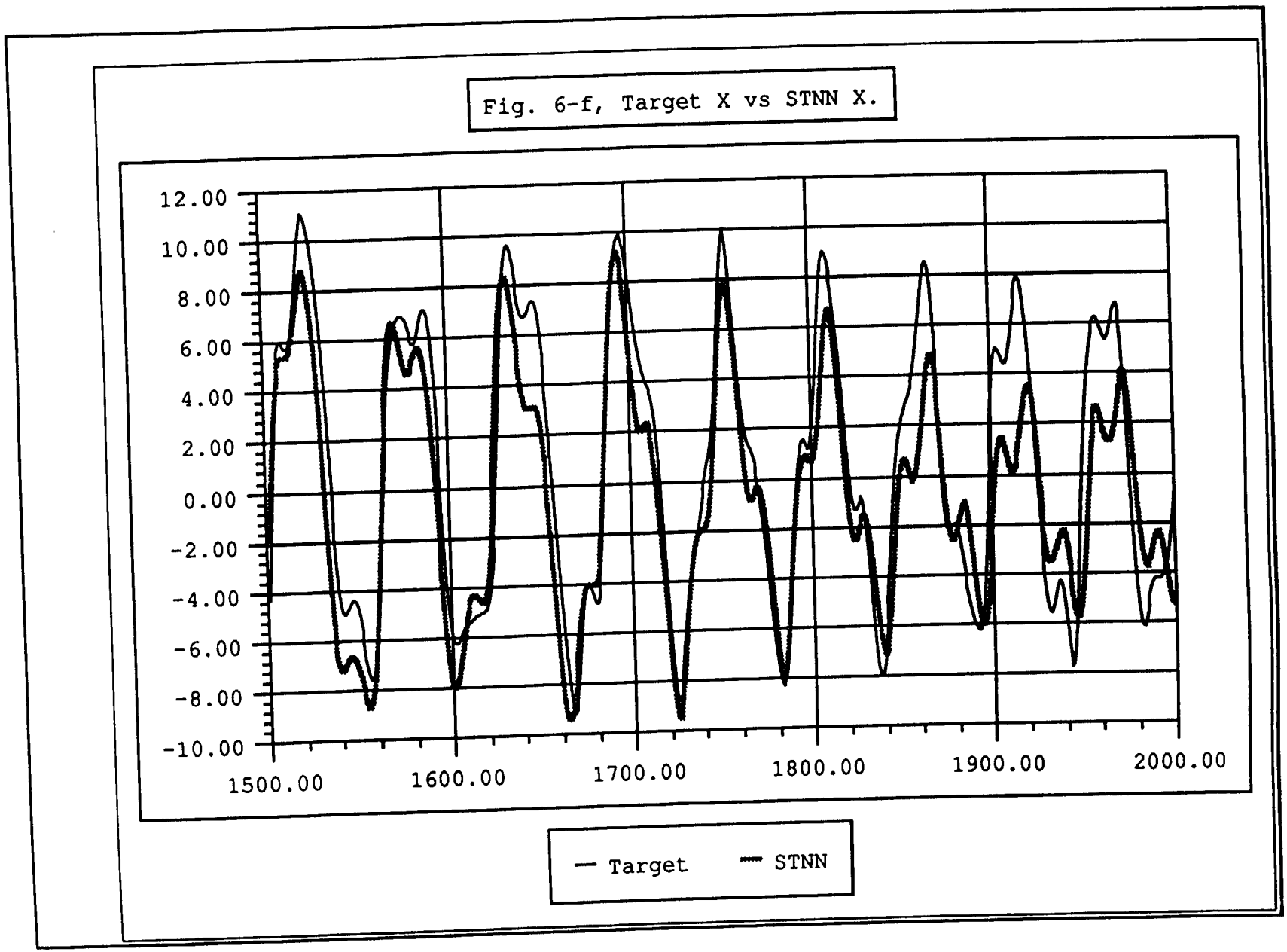


Fig. 6-f, Target X vs STNN X.
12.00
10.00
8.00
6.00
4.00
2.00
0.00
-2.00
-4.00
-6.00
-8.00
-10.00
1500.00
1600.00
1700.00
1800.00
1900.00
2000.00
Target
STNN

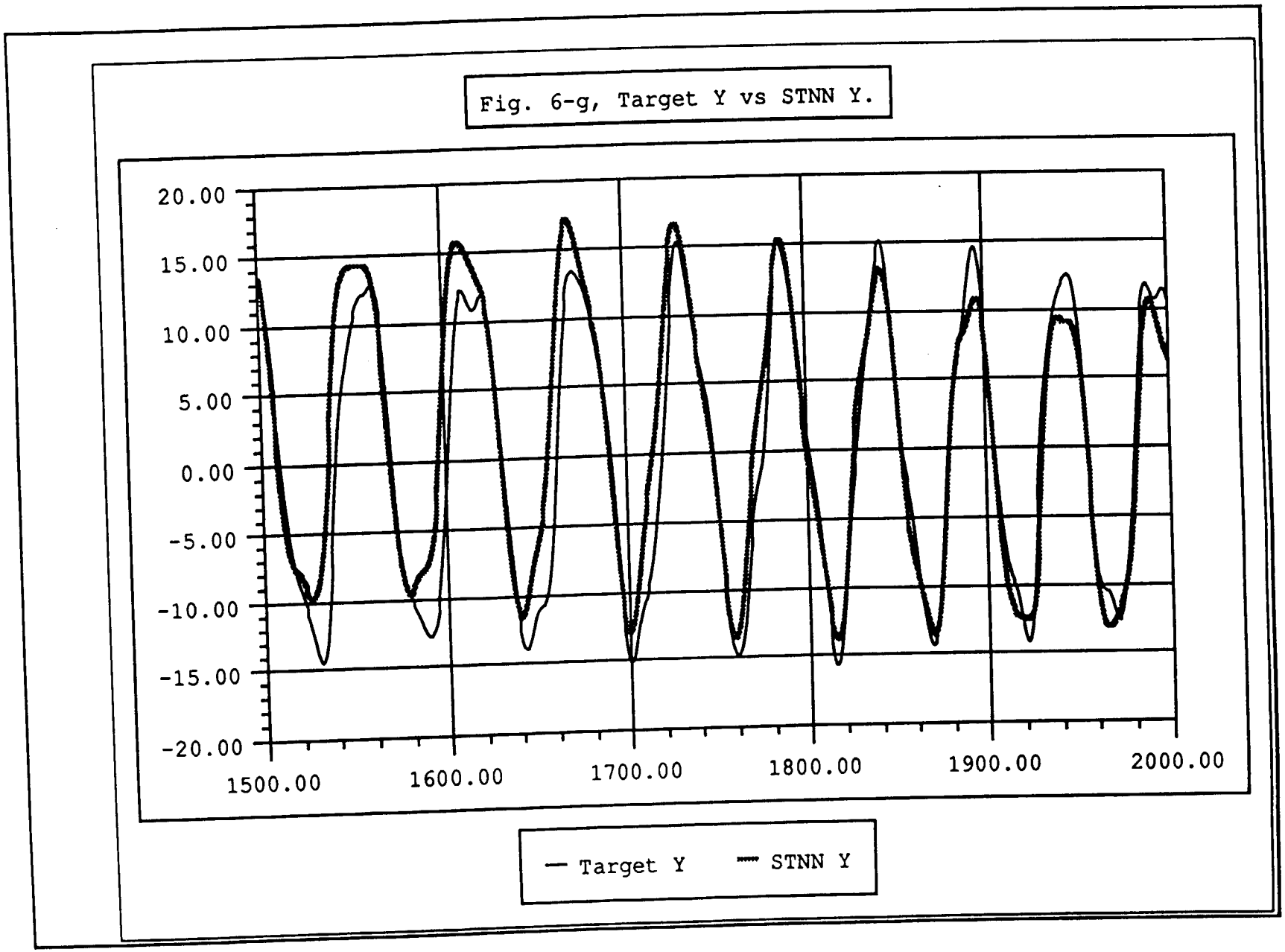


Fig. 6-g, Target Y vs STNN Y.
20.00
15.00
10.00
5.00
0.00
-5.00
-10.00
-15.00
-20.00
1500.00
1600.00
1700.00
1800.00
1900.00
2000.00
Target Y
STNN Y

C-3

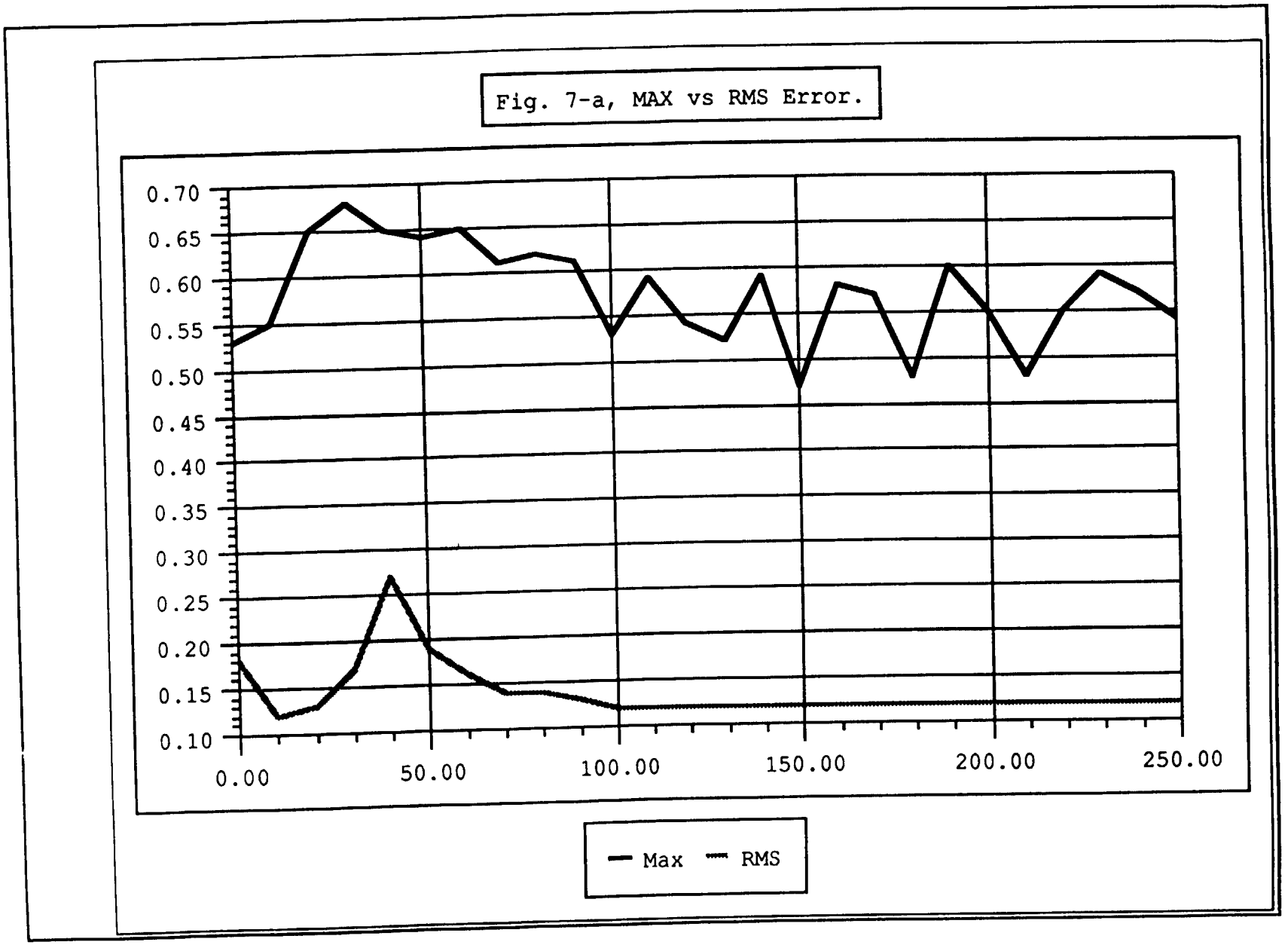


Fig. 7-a, MAX vs RMS Error.
0.70
0.65
0.60
0.55
0.50
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.00
50.00
100.00
150.00
200.00
250.00
Max
RMS

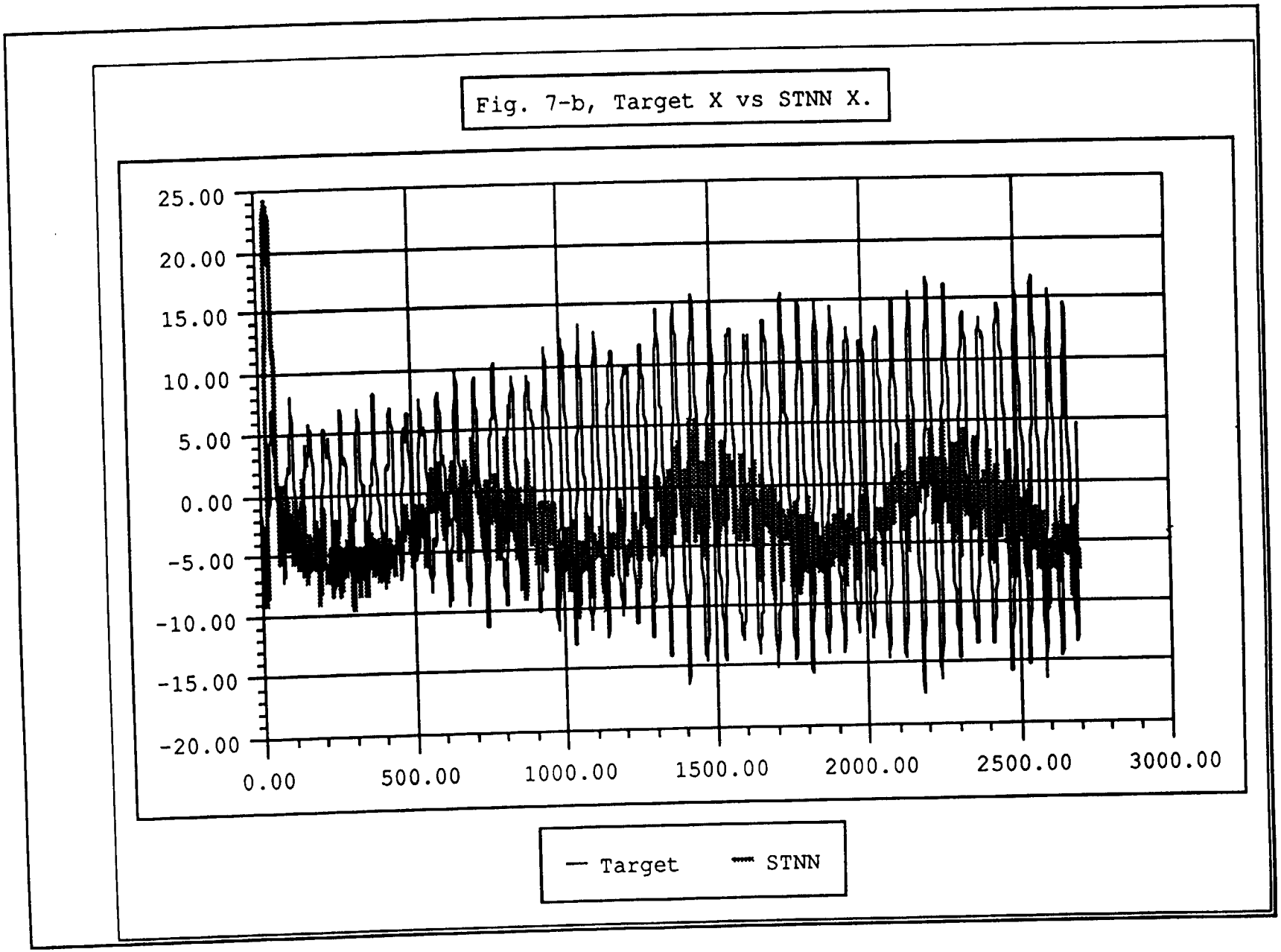


Fig. 7-b, Target X vs STNN X.
25.00
20.00
15.00
10.00
5.00
0.00
-5.00
-10.00
-15.00
-20.00
0.00
500.00
1000.00
1500.00
2000.00
2500.00
3000.00
Target
STNN

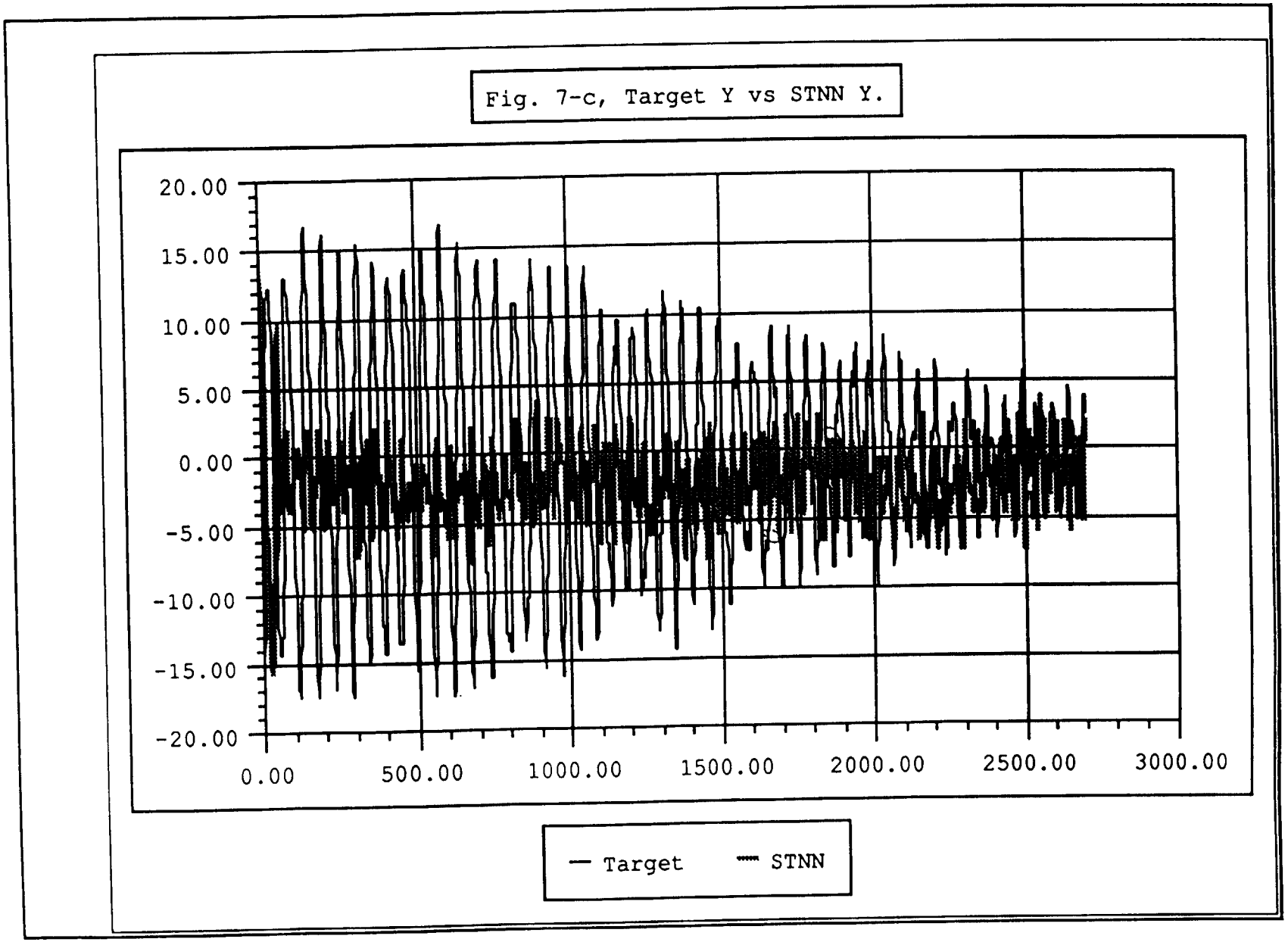


Fig. 7-c, Target Y vs STNN Y.
20.00
15.00
10.00
5.00
0.00
-5.00
-10.00
-15.00
-20.00
0.00
500.00
1000.00
1500.00
2000.00
2500.00
3000.00
Target
STNN

compare its predictions based on one bead model. The filter's gains are adjusted until the difference between simulated and actual rate gyro data is minimized. At this time the filter can be used to provide the crew with an estimation of skiprope motion parameters so that appropriate orbital maneuvers can be carried out to damp the skiprope motion.

The time-domain Kalman filter technique runs at real time on an HP 9000 computer and is expected to run faster than real time on the "back-room" computers at the Johnson Space Center. However, depending on the tether length, skiprope frequency, and actual phase angle during the mission, this single bead simulation could take from several seconds to several minutes to reasonably estimate the skiprope motion. During various phases of the mission, including satellite spin and current pulsing activities, the filter requires the maximum amount of time synchronized data to arrive at a prediction. In addition, the filter uses only a one bead simulation of the tethered system, and therefore, the predictions may not be as accurate as a multi bead simulation. Verification of the time-domain skiprope observer must be performed on Multi Purpose Support Room (MPSR) hardware to verify that the observer will perform within the required limits. As of this writing, the overall level of confidence that the time-domain skiprope observer will perform adequately in real time situations is not high. However, the technique is very well-known, well-studied and frequently used in space operations and thus there are no questions about the validity of the technique.

A frequency based method for skiprope observation has also been studied. This technique requires three full cycles, or up to 1500 seconds, of downlinked data that relates skiprope activity to arrive at a prediction. This method works well during steady state conditions but is less effective with perturbations such as satellite spinning or current pulsing. The frequency based method is also susceptible to data dropout and rate gyro saturation. The frequency domain filter is designed specifically to support the yaw maneuver scheduled during On-station 2 and does not perform well at On-station1. Results indicate that the frequency domain filter performs well for a 50 m skiprope, but not for skiprope in the 10 m to 15 m range. Again, the method is well-known, well-studied and has a history of utilization in many applications.

The STNN method that we have proposed offers the advantage that it is trained using a high fidelity simulation where from 10 to 50 beads are modelled, and the orbital environment is also modelled with high fidelity and accuracy. In addition, the network can be trained to account for a changing orbital environment due to crew inputs. The Orbital Operations Simulator (OOS) used to evaluate the STNN skiprope observer is also used for actual crew training during tethered satellite deployment and retrieval. Crew inputs to maintain attitude and damp skiprope motion may be logged and included in STNN training data. STNN is based on the promise that it can be trained for nonlinear behavior and it will perform proper interpolation for this non-linearity. Our objective is to demonstrate that the STNN skiprope observer can accurately predict skiprope parameters more accurately and with fewer data cycles than either the time-domain or frequency-domain methods.

Disadvantages of STNN based skiprope observer are many, especially in light of well-known methods. First of all, this a new method, and therefore, is not well-known. It has not been applied earlier in any other application and therefore it does not have a history like frequency domain method. There is no rigorous mathematical proof that neural networks map one set of parameters to another set of parameters uniquely. Thus, the method may not provide a confidence required for space operations. Further, the verification and validation of this method has to be carried out in detail. This task will be resource consuming and may prohibit the application of the method to real operations.

6.0 Future Work

Since the time-domain method has been baselined as the skiprope observer for TSS-1, the STNN skiprope observer will probably not be used operationally during the mission. However, we plan to utilize the STL's capability to receive telemetry data to test the STNN skiprope observer during the TSS-1 mission to evaluate it's performance. If the STNN observer meets the requirements, then it may be used on follow-up missions that are currently being proposed.

We plan to generate training data sets using the OOS that has a very high fidelity bead model for the tether dynamics and high fidelity space shuttle and Italian Satellite models with respective control systems. We can simulate up to 50 beads for the tether behavior and generate required data for the satellite attitude and skiprope parameters. Training and test data sets have already been prepared using the OST2 segment simulation.

Our next step is to configure STNN and train it using the data set. Once the training is completed, we will test the performance of the STNN using part of the data. Based on the results we will enhance the STNN configuration and perform retraining if necessary. Using the TSS-1 mission profile, we will generate the skip rope data for the On station 1, retrieval up to 2.4 km. and Onstation-2 and final retrieval phases so that we can train the STNN for full retrieval phase. We will test the performance of STNN using simulated telemetry data (while connecting the simulation with STNN) and see if the STNN can perform real time.

References

1. Coledan, S. : "Tethered Satellite Advances", Space News, vol. 2, no. 15, p. 8, 1991.
2. Powers, C.B., Shea, C., and McMahan, T. : "The First Mission of the Tethered Satellite System", A special brochure developed by the Tethered Satellite System Project Office, NASA/Marshall Space Flight Center, Huntsville, Alabama, U.S. GPO 1992-324-999, 1992.
3. Ioup, G.E., Ioup, J.W., Rodrique, S.M., Amini, A.M., Rayborn, G.H., and Carroll, S. : "Frequency Domain Skiprope Observer", Skiprope Containment Status Meeting held at Denver, Sep. 10-11, 1991. (Research supported by NASA Contract NA8-38841)
4. Glowczwski, R. : "Time Domain Skiprope Observer Overview", Skiprope Containment Status Meeting held at Martin Marietta, Denver, Sep. 10-11, 1991.
5. Klir, G.J., and Folger, T.A. : "Fuzzy Sets, Uncertainty, and Information", Prentice-Hall, New Jersey, 1988.
6. Kosko, B. : "Neural Networks and Fuzzy Systems", Prentice-Hall, New Jersey, 1992.
7. Freeman, J.A., and Skapura, D.M. : "Neural Networks, Algorithms, Applications, and Programming Techniques", Addition-Wesley, 1991.
8. Lea, R.N., and Jani, Y. : " Applications of Fuzzy Logic to Control and Decision Making", Technology 2000 Proceedings, NASA Conference Publication 3109, Vol. 2, p. 67, 1990.
9. Lea, R.N., Hoblit, J., and Jani, Y. : "Performance Comparison of a Fuzzy Logic based Attitude Controller with the Shuttle On-orbit Digital Auto Pilot", North American Fuzzy Information Processing Society 1991 Workshop Proceedings, pp 291-295, 1991.
10. Lea, R.N., Villarreal, J., Jani, Y., and Copeland, C. : "Fuzzy Logic Based Tether Control", North American Fuzzy Information Processing Society 1991 Workshop Proceedings, pp 398-402, 1991.
11. Villarreal, J.A., and Shelton, R.O. : "A Space-Time Neural Network", International Journal of Approximate Reasoning, vol. 6, number 2, February 1992.

Structure Identification in Fuzzy Inference Using Reinforcement Learning

Hamid R. Berenji[1] and Pratap Khedkar[2]
AI Research Branch, MS: 269-2
NASA Ames Research Center
Mountain View, CA 94035

[1]Sterling Software, berenji@ptolemy.arc.nasa.gov
[2]EECS Department, University of California, Berkeley, CA 94720, khedkar@cs.berkeley.edu

In our previous work on the GARIC architecture, we have shown that the system can start with the surface structure of the knowledge base (i.e., the linguistic expression of the rules) and learn the deep structure (i.e., the fuzzy membership functions of the labels used in the rules) by using reinforcement learning. Assuming the surface structure, GARIC refines the fuzzy membership functions used in the consequents of the rules using a gradient descent procedure. This hybrid fuzzy logic and reinforcement learning approach can learn to balance a cart-pole system and to backup a truck to its docking location after a few trials.

In this paper, we discuss how to do structure identification using reinforcement learning in fuzzy inference systems. This involves identifying both surface as well as deep structure of the knowledge base. The term set of fuzzy linguistic labels used in describing the values of each control variable must be derived. In this process, splitting a label refers to creating new labels which are more granular than the original label and merging two labels creates a more general label. Splitting and merging of labels directly transform the structure of the action selection network used in GARIC by increasing or decreasing the number of hidden layer nodes.

After each splitting or merging of a label, the learning resumes by refining the fuzzy membership functions used in the consequent of the rules. Depending on the performance of the learning algorithm after a change in the structure of the system, our algorithm selects the next node(s) to be split or to be merged and the process is then iterated. The proposed method provides a more flexible structure for encoding the prior control knowledge where both the structure of the rules and the fuzzy membership functions used in the labels can be learned automatically.

Approximation Paper: Part I

J. J. Buckley
Mathematics Department
University of Alabama at Birmingham
Birmingham, AL 35294

Abstract: In this paper we discuss approximations between neural nets, fuzzy expert systems, fuzzy controllers, and continuous processes.

1. Introduction.

In this section we will first present the definition of a continuous process (system). The following sections discuss neural networks, fuzzy expert systems, the fuzzy controller, and approximations between all four objects. The last section has a brief summary and suggestions for future research.

A system (process) S has m inputs x_i and n outputs y_j. Let $x = (x_1, \cdots, x_m)$ and $y = (y_1, \cdots, y_n)$. The inputs are all bounded so assume that each input is scaled to belong to $[0, 1]$. This means that S will be a mapping from $[0, 1]^m$ into R^n written as $y = S(x)$. We assume that S is continuous and let $\mathcal{S}$ denote the set of all continuous mappings from $[0, 1]^m$ into R^n. By a continuous process (system) we will mean any S in $\mathcal{S}$.

2. Neural Nets.

The neural network will be a layered, feedforward, net with m input neurons and n output neurons. The net can have any number of hidden layers. Input to the net will be a vector $x = (x_1, \cdots, x_m)$, x_i in $[0, 1]$ all i, and the output is also a vector $y = (y_1, \cdots y_n)$. We assume that the activation function[1] within a neuron is continuous. Therefore, the neural net is a continuous mapping from input x in $[0, 1]^m$ to output y in R^n denoted as $y = F(x)$. We note that F belongs to $\mathcal{S}$.

The following result comes from recent publications in the neural network literature ([1], [9], [14], [15], [16]) where it was shown that multilayer feedforward nets are universal approximators. Given S in $\mathcal{S}$ and $\epsilon > 0$ there is a neural net F so that $|S(x) - F(x)| < \epsilon$ for all x in $[0, 1]^m$, see [8].

3. Fuzzy Expert Systems.

The fuzzy expert system will contain one block of rules written as[2]

$$\mathcal{R}_i : \text{ If } X = \bar{A}_i, \text{ then } Z = \bar{C}_i,$$

$1 \leq i \leq N$. $\bar{A}_i$ and $\bar{C}_i$ represent fuzzy subsets of the real numbers. If $\bar{A}$ denotes any fuzzy subset of the reals, then $\bar{A}(x)$ is its membership function evaluated at x.

Let $X = \bar{A}'$ be the input to the fuzzy expert system. The rules are evaluated using some method of approximate reasoning (fuzzy logic) producing final conclusion (output) $Z = \bar{C}'$. Let $\mathcal{A}$ denote the type of approximate reasoning employed by the fuzzy expert system.

We now discretize all the fuzzy sets. Let $x_0, \cdots, x_{N_1}$ be numbers covering the support of all the $\bar{A}_i$ and $\bar{A}'$ and let $z_0, \cdots, z_{N_2}$ be numbers covering the support of all the $\bar{C}_i$ and $\bar{C}'$. Let $x = \left(\bar{A}'(x_0), \cdots, \bar{A}'(x_{N_1})\right)$ in $[0,1]^m$ if $m = N_1 + 1$ and let $y = \left(\bar{C}'(z_0), \cdots, \bar{C}'(z_{N_2})\right)$ in R^n if $n = N_2 + 1$. Then x is the input to the fuzzy expert system and y is its output. So, the fuzzy expert system is a mapping from x in $[0,1]^m$ to y in R^n which we write as $y = G(x)$. We assume that we have selected an $\mathcal{A}$ so that this mapping is continuous. Hence, G also belongs to $\mathcal{S}$.

The first papers discussing the approximation of a neural net by a fuzzy expert system were ([7], [10]) but the main result was proven in [8]. Given a neural net F and $\epsilon > 0$ there exists a fuzzy expert system (block of rules and $\mathcal{A}$) so that $|F(x) - G(x)| < \epsilon$ for all x in $[0,1]^m$. In [8] we found only one $\mathcal{A}$ that will do the job. From the second section we may conclude that given any S in $\mathcal{S}$ and $\epsilon > 0$ there is a fuzzy expert system G so that $|S(x) - G(x)| < \epsilon$ for all x in $[0,1]^m$.

4. Fuzzy Controller.

It will be easier now if we restrict $m = 2$ and $n = 1$, however we can generalize to other values of m and n. Let us assume that the fuzzy controller has only two inputs error = e and change in error = Δe, and only one defuzzified output δ. We assume that the inputs have been scaled to lie in $[0,1]$. The fuzzy control rules are of the form

$$\begin{aligned}\mathcal{R} : &\text{If Error } = \bar{A}_i \text{ and Change in Error } = \bar{B}_j,\\ &\text{then Control } = \bar{C}_k.\end{aligned}$$

Once a method of evaluating the rules has been chosen and a procedure for defuzzification is adopted, the fuzzy controller is a mapping H from $(e, \Delta e)$ in $[0,1]^2$ to δ in R. We assume that the internal operations within the controller are continuous so that H belongs to $\mathcal{S}$ for $m = 2$, $n = 1$. In general, we can have H in $\mathcal{S}$ for any m and n.

Different types of fuzzy controllers are discussed in ([2], [3]). In [5] and [6] we identified two types of fuzzy controller, now labeled $\mathcal{T}_1$ and $\mathcal{T}_2$, that can approximate any S in $\mathcal{S}$ to any degree of accuracy. A different type of approximation result of S in $\mathcal{S}$, by fuzzy controllers, is presented in [4]. Let this third type of controller be called $\mathcal{T}_3$. So, given S in $\mathcal{S}^3$, $\epsilon > 0$ and i in $\{1,2,3\}$, there is a fuzzy controller H in $\mathcal{T}_i$ so that $|S(x) - H(x)| < \epsilon$ for all x in $[0,1]^2$. Hence, from the previous two sections, we may approximate fuzzy expert systems and neural nets, to any degree of accuracy, by fuzzy controllers.

5. Conclusions.

The results discussed in this paper may be summarized as follows: given any two objects E_1 and E_2 from the set {continuous process, neural net, fuzzy expert system, fuzzy controller}, we can use an E_1 to approximate an E_2 to any degree of accuracy. Assumptions needed to obtain this result are discussed within the paper.

Future research is needed to extend these results in many directions including: (1) fuzzy neural nets ([11], [12]); (2) neural nets that employ t-norms and t-conorms to process information [13]; (3) finding more fuzzy expert systems ($\mathcal{A}$'s) that can be used to approximate neural nets; and (4) discovering other types of fuzzy controllers that approximate continuous systems.

6. References.

1. E. K. Blum and L. K. Li, Approximation Theory and Feedforward Networks, Neural Networks 3 (1991) 511–515.
2. J. J. Buckley, Theory of the Fuzzy Controller: An Introduction, Fuzzy Sets and Systems. To appear.
3. J. J. Buckley, Theory of the Fuzzy Controller: A Brief Survey, in: C. V. Negoita (ed.), Handbook of Cybernetics and Systems, Marcel Dekker, N. Y.. To appear.

4. J. J. Buckley, Controllable Processes and the Fuzzy Controller, Fuzzy Sets and Systems. Submitted.
5. J. J. Buckley, Sugeno Type Controllers are Universal Controllers, Fuzzy Sets and Systems. Submitted.
6. J. J. Buckley, Universal Fuzzy Controllers, Automatica. Submitted.
7. J. J. Buckley and E. Czogala, Fuzzy Models, Fuzzy Controllers and Neural Nets, Frontiers of Applied Mathematics, SIAM. To appear.
8. J. J. Buckley, Y. Hayashi and E. Czogala, On the Equivalence of Neural Nets and Fuzzy Expert Systems, Fuzzy Sets and Systems. Submitted.
9. G. Cybenko, Approximation by Superpositions of a Sigmoidal Function, Math. of Control, Signals, and Systems 2 (1989) 303–314.
10. Y. Hayashi, J. J. Buckley and E. Czogala, Fuzzy Expert Systems Versus Neural Networks, Int. J. Approximate Reasoning. Submitted.
11. Y. Hayashi, J. J. Buckley and E. Czogala, Fuzzy Neural Network with Fuzzy Signals and Weights, Int. J. Intelligent Systems. Submitted.
12. Y. Hayashi, J. J. Buckley and E. Czogala, Systems Engineering Applications of Fuzzy Neural Networks, J. of Systems Engineering. Submitted.
13. Y. Hayashi, E. Czogala and J. J. Buckley, Fuzzy Max-Min Neural Controller, Proceedings FUZZ IEEE, San Diego, March, 1992. To appear.
14. K. Hornik, Approximation Capabilities of Multilayer Feedforward Networks, Neural Networks 4 (1991) 251–257.
15. K. Hornik, M. Stinchcombe, and H. White, Multilayer Feedforward Networks are Universal Approximators, Neural Networks 2 (1989) 359–366.
16. V. Y. Kreinovich, Arbitrary Nonlinearity is Sufficient to Represent all Functions by Neural Networks: A Theorem, Neural Networks 4 (1991) 381–383.

7. Notes

[1] In general, we assume that the mapping from input to output, for any neuron in the net, is a continuous operation.

[2] We could consider more complicated rules and/or more blocks of rules.

[3] $m = 2$ and $n = 1$. Can generalize.

NONLINEAR RESCALING OF CONTROL VALUES SIMPLIFIES FUZZY CONTROL

H. VanLangingham[1], A. Tsoukkas[1], V. Kreinovich[2], C. Quintana[2]
[1]The Bradley Department of Electrical Engineering, Virginia Polytechnic Institute and State University, Blacksburg VA 24061, USA
[2]Computer Science Department, University of Texas at El Paso, El Paso, TX 79968, USA

Abstract. Traditional control theory is well-developed mainly for linear control situations. In non-linear cases there is no general method of generating a good control, so we have to rely on the ability of the experts (operators) to control them. If we want to automate their control, we must acquire their knowledge and translate it into a precise control strategy.

The experts' knowledge is usually represented in non-numeric terms, namely, in terms of uncertain statements of the type "if the obstacle is straight ahead, the distance to it is small, and the velocity of the car is medium, press the brakes hard". Fuzzy control is a methodology that translates such statements into precise formulas for control. The necessary first step of this strategy consists of assigning membership functions to all the terms that the expert uses in his rules (in our sample phrase these words are "small", "medium", and "hard").

The appropriate choice of a membership function can drastically improve the quality of a fuzzy control. In the simplest cases, we can take the functions whose domains have equally spaced endpoints. Because of that, many software packages for fuzzy control are based on this choice of membership functions. This choice is not very efficient in more complicated cases. Therefore, methods have been developed that use neural networks or genetic algorithms to "tune" membership functions. But this tuning takes lots of time (for example, several thousands iterations are typical for neural networks).

In some cases there are evident physical reasons why equally spaced domains do not work: e.g., if the control variable u is always positive (i.e., if we control temperature in a reactor), then negative values (that are generated by equal spacing) simply make no sense. In this case *it sounds reasonable to choose another scale* $u' = f(u)$ *to represent* u, *so that equal spacing will work fine for* u'.

In the present paper we formulate the problem of finding the best rescaling function, solve this problem, and show (on a real-life example) that after an optimal rescaling, the un-tuned fuzzy control can be as good as the best state-of-art traditional non-linear controls.

1. INTRODUCTION TO THE PROBLEM

Traditional control theory is not always applicable, so we have to use fuzzy control. Traditional control theory is well-developed mainly for linear control situations. In non-linear cases, although for many cases there are good recipes, there is still no general method of generating a good control (see, e.g., [M91]).

Therefore, we have to rely on the ability of the experts (operators) to control these systems. If we want to automate their control, we must acquire transform their knowledge it into a precise control strategy.

The experts' knowledge is usually represented in non-numeric terms, namely, in terms of uncertain statements of the type "if the obstacle is straight ahead, the distance to it is small, and the velocity of the car is medium, press the brakes hard". Fuzzy control is a methodology that

translates such statements into precise formulas for control. Fuzzy control was started by L. Zadeh and E. H. Mamdani [Z71], [CZ72], [Z73], [M74] in the framework of fuzzy set theory [Z65]. For the current state of fuzzy control the reader is referred to the surveys [S85], [L90] and [B91].

Choice of membership functions: an important first step of fuzzy control methodology. The necessary first step of this methodology consists of assigning membership functions to all the terms that the expert uses in his rules (in our sample phrase these words are "small", "medium", and "hard"). The appropriate choice of a membership function can drastically improve the quality of a fuzzy control.

Simplest case: equally spaced functions. In the simplest cases, we can take the functions whose domains have equally spaced endpoints: e.g., we can fix a neutral value N (usually, $N = 0$), and a number Δ, and take "negligible" with the domain $[N - \Delta, N + \Delta]$, "small positive" with the domain $[N, N + 2\Delta]$, "medium positive" with the domain $[N + \Delta, N + 3\Delta]$, etc. Correspondingly, "small negative has the domain $[N - 2\Delta, N]$, "medium negative corresponds to the domain $[N - 3\Delta, N - \Delta]$, etc. If an interval $[a - \Delta, a + \Delta]$ is given, then we can take a membership function $\mu(x)$ that is equal to 0 outside this interval, equal to 1 for $x = a$, and is linear on the intervals $[a - \Delta]$ and $[a, a + \Delta]$. Many software packages for fuzzy control are based on this choice of membership functions.

What is usually done in more complicated cases. This choice of equally spaced functions is not always very efficient in more complicated cases. Therefore, methods have been developed that use neural networks or genetic algorithms to "tune" membership functions (see, e.g., numerous papers in [RSW92]). But this tuning takes lots of time (for example, several thousands iterations are typical for neural networks).

The idea of a rescaling. In some cases there are evident physical reasons why equally spaced domains do not work. For example, if the control variable u is always positive (i.e., if we control the flow of some substance into a reactor), then negative values (that will be eventually generated by an equal spacing method) simply make no sense.

A natural idea is to choose another scale $u' = f(u)$ to represent the control variable u, so that equal spacing will work fine for u'. This idea is in good accordance with our common-sense description of physical processes. For example, from the physical viewpoint it is quite possible to describe the strength of an earthquake by its energy, but, when we talk about its consequences, it is much more convenient to use a logarithmic scale (called Richter scale). Non-linear scales are used to describe amplifiers and noise (decibels, in electrical engineering), to describe hardness of different minerals in geosciences, etc. (for a general survey of different scales and rescalings see [SKLT71, 89]).

In our case we want to design such a scale that for $f(u)$ the equally spaced endpoints $N - k\Delta$ and $N + k\Delta$ would make sense for all integers k. Therefore, we are looking for a function $f(u)$, whose domain is the set of all positive values, and whose range is all possible real numbers. In mathematical notations, f must map $(0, \infty)$ onto $(-\infty, \infty)$. There are lots of such functions, and evidently not all of them will improve the control. So we arrive at the following problem:

The main problem. *What rescaling to choose?*

What we are planning to do. We formulate the problem of choosing the best rescaling function $f(u)$ as a mathematical optimization problem, and then we solve this problem under some reasonable optimality criteria. As a result, we get an optimal function $f(u)$. We show that its application to non-linear systems really improves fuzzy control.

2. MOTIVATIONS OF THE PROPOSED MATHEMATICAL DEFINITIONS

Why is this problem difficult? We want to find a scaling function $f(u)$ that is the best in some reasonable sense, that is, for which some characteristic I attains the value that corresponds to the best performance of the resulting fuzzy control. As examples of such characteristics, we can take an average running time of an algorithm, or some characteristics of smoothness or stability of the resulting control, etc. The problem is that even for the simplest linear plants (controlled systems), we do not know how to compute any of these possible characteristics. How can we find $f(u)$ for which $I(f(u))$ is optimal if we cannot compute $I(f(u))$ even for a single function $f(u)$? There does not seem to be a likely answer.

However, we will show that this problem is solvable (and give the solution).

The basic idea for solving these kind of problems is described in [K90]; for its application to fuzzy logic see [KK90], to neural networks see [KQ91], to genetic algorithms see [KQF92], and to different problems of fuzzy control see [KQLFLKBR92].

We must choose a family of functions, not a single function. Suppose that for some physical quantity u (e.g., for x coordinate) equal spacing leads to a reasonably good control strategy.

In order to get numerical values of x coordinate, we must fix some starting point and some measuring unit (e.g., a meter). In principle we could as well choose feet to describe length. Then the numerical values of all the coordinates will be different (x meters are equal to λx feet, where λ is the number of feet in 1 meter). However, the intervals that were equally spaced when we used one unit, are still equally spaced, if we use another unit to measure this coordinate.

In a similar way, we could choose a different starting point for the x coordinate. If we take as a starting point a point that had a coordinate x_0 (so that now its coordinate is 0), then all other coordinates will be shifted: $x \to x - x_0$. Again intervals that were equal in the old scale (x) will still be equal if we measure then in the new scale ($x - x_0$).

We can also change both the measuring unit and the starting point. This way we arrive at a transformation $x \to \lambda x + x_0$.

Summarizing: if x is a reasonable scale, in the sense that equally spaced membership functions lead to a reasonably good control, then the same is true for any scale $\lambda x + x_0$, where $\lambda > 0$, and x_0 is any real number. The reason is that if we have a sequence of equally spaced intervals $[N + k\Delta, N + (k+1)\Delta]$, then these intervals will remain equally spaces after these linear rescalings $x \to \lambda x + x_0$: namely, these intervals will turn into intervals $[N' + k\Delta', N' + (k+1)\Delta']$, where $N' = \lambda N + x_0$ and $\Delta' = \lambda\Delta$.

Let us now consider a scale u, for which equal spacing does not work. Assume that $u \to f(u)$ is a transformation, after which equal spacing becomes applicable. This means that if we use $f(u)$ as a new scale, then equal spacings work fine. But as we have just shown, for any $\lambda > 0$ and x_0 equal spacing will also work fine for the scale $\lambda f(u) + x_0$.

Therefore, if $f(u)$ is a function that transforms the initial scale into a scale, for which equal spacing works fine, then for every $\lambda > 0$ and x_0 the function $f'(u) = \lambda f(u) + x_0$ has the same desired property.

This means that there is no way to pick one function $f(u)$, because with any function $f(u)$, the whole family of functions $\lambda f(u) + x_0$ has the same property. Therefore, desired functions form

a *family* $\{\lambda f(u) + x_0\}_{\lambda>0, x_0}$. Hence, instead of choosing a single function, we must formulate a problem of choosing a family.

Which family is the best? Among all such families, we want to choose the best one. In formalizing what "the best" means, we follow the general idea outlined in [K90] and applied to neural networks in [KQ91]. The criteria to choose may be computational simplicity, stability or smoothness of the resulting control, etc. In mathematical optimization problems, numeric criteria are most frequently used, where to every family we assign some value expressing its performance, and choose a family for which this value is maximal. However, it is not necessary to restrict ourselves to such numeric criteria only. For example, if we have several different families that lead to the same average stability characteristics T, we can choose between them the one that leads to the maximal smoothness characteristics P. In this case, the actual criterion that we use to compare two families is not numeric, but more complicated: *a family Φ_1 is better than the family Φ_2 if and only if either $T(\Phi_1) < T(\Phi_2)$, or $T(\Phi_1) = T(\Phi_2)$ and $P(\Phi_1) < P(\Phi_2)$*. A criterion can be even more complicated. What a criterion *must* do is to allow us for every pair of families to tell whether the first family is better with respect to this criterion (we'll denote it by $\Phi_2 < \Phi_1$), or the second is better ($\Phi_1 < \Phi_2$) or these families have the same quality in the sense of this criterion (we'll denote it by $\Phi_1 \sim \Phi_2$).

The criterion for choosing the best family must be consistent. Of course, it is necessary to demand that these choices be consistent, e.g., if $\Phi_1 < \Phi_2$ and $\Phi_2 < \Phi_3$ then $\Phi_1 < \Phi_3$.

The criterion must be final. Another natural demand is that this criterion must be *final* in the sense that it must choose a *unique* optimal family (i.e., a family that is better with respect to this criterion than any other family).

The reason for this demand is very simple. If a criterion does not choose any family at all, then it is of no use. If several different families are "the best" according to this criterion, then we still have a problem choosing the absolute "best" family. Therefore, we need some additional criterion for that choice. For example, if several families turn out to have the same stability characteristics, we can choose among them a family with maximal smoothness. So what we actually do in this case is abandon that criterion for which there were several "best" families, and consider a new "composite" criterion instead: Φ_1 is better than Φ_2 according to this new criterion if either it was better according to the old criterion, or according to the old criterion they had the same quality, and Φ_1 is better than Φ_2 according to the additional criterion. In other words, if a criterion does not allow us to choose a unique best family, it means that this criterion is not ultimate; we have to modify it until we come to a final criterion that will have that property.

The criterion must be reasonably invariant. We have already discussed the effect of changing units in a new scale $f(u)$. But it is also possible to change units in the original scale, in which the control u is described. If we use a unit that is c times smaller, then a control whose numeric value in the original scale was u, will now have the numeric value cu. For example, if we initially measured the flux of a substance (e.g., rocket fuel) into the reactor by kg/sec, we can now switch to lb/sec.

Comment. There is no physical sense in changing the starting point for u, because we consider the control variable that takes only positive values, and so 0 is a fixed value, corresponding to the minimal possible control.

We are looking for the universal rescaling method, that will be applicable to any reasonable situation (we do not want it to be adjustable to the situation, because the whole purpose of this rescaling is to avoid time-consuming adjustments). Suppose now that we first used kg/sec,

compared two different scaling functions $f(u)$ and $\tilde{f}(u)$, and it turned out that $f(u)$ is better (or, to be more precise, that the family $\Phi = \{\lambda f(u) + x_0\}$ is better than the family $\tilde{\Phi} = \{\lambda \tilde{f}(u) + x_0\}$). It sounds reasonable to expect that the relative quality of the two scaling functions should not depend on what units we used for u. So we expect that when we apply the same methods, but with the values of control expressed in lb/sec, then the results of applying $f(u)$ will still be better than the results of applying $\tilde{f}(u)$. But the result of applying the function $f(u)$ to the control in lb/sec can be expressed in old units (kg/sec) as $f(cu)$, where c is a ratio of these two units. So the result of applying the rescaling function $f(u)$ to the data in new units (lb/sec) coincides with the result of applying a new scaling function $f_c(u) = f(cu)$ to the control in old units (kg/sec). So we conclude that if $f(u)$ is better than $\tilde{f}(u)$, then $f_c(u)$ must be better than $\tilde{f}_c(u)$, where $f_c(u) = f(cu)$ and $\tilde{f}_c(u) = \tilde{f}(cu)$. This must be true for every c because we could use not only kg/sec or lb/sec, but arbitrary units as well.

Now we are ready for the formal definitions.

3. DEFINITIONS AND THE MAIN RESULT

Definitions. By a *rescaling function* (or a *rescaling* for short), we mean a strictly monotonic function that maps the set of all positive real numbers $(0, \infty)$ onto the set of all real numbers $(-\infty, +\infty)$. We say that two rescalings $f(u)$ and $f'(u)$ are *equivalent* if $f'(u) = Cf(u) + x_0$ for some positive constant C and for some real number x_0.

Comment. As we have already mentioned, if we apply two equivalent rescalings, we will get two scales that are either both leading to a good control, or are both inadequate.

By a *family* we mean the set of functions $\{Cf(u) + x_0\}$, where $f(u)$ is a fixed rescaling, C runs over all positive real numbers, and x_0 runs over all real numbers. The set of all families will be denoted by S.

A pair of relations $(<, \sim)$ is called *consistent* [K90], [KK90], [KQ91] if it satisfies the following conditions:
(1) if $F < G$ and $G < H$ then $F < H$;
(2) $F \sim F$;
(3) if $F \sim G$ then $G \sim F$;
(4) if $F \sim G$ and $G \sim H$ then $F \sim H$;
(5) if $F < G$ and $G \sim H$ then $F < H$;
(6) if $F \sim G$ and $G < H$ then $F < H$;
(7) if $F < G$ then it is not true that $G < F$ or $F \sim G$.

Assume a set A is given. Its elements will be called *alternatives*. By an *optimality criterion* we mean a consistent pair $(<, \sim)$ of relations on the set A of all alternatives. If $G < F$, we say that F is *better* than G; if $F \sim G$, we say that the alternatives F and G are *equivalent* with respect to this criterion. We say that an alternative F is *optimal* (or *best*) with respect to a criterion $(<, \sim)$ if for every other alternative G either $G < F$ or $F \sim G$.

We say that a criterion is *final* if there exists an optimal alternative, and this optimal alternative is unique.

Comment. In the present paper we consider optimality criteria on the set S of all families.

Definitions. By a *result of a unit change* in a function $f(u)$ to a unit that is $c > 0$ times smaller we mean a function $f_c(u) = f(cu)$. By the *result of a unit change* in a family Φ by $c > 0$ we mean the set of all the functions that are obtained by this unit change from $f \in \Phi$. This result will be

denoted by $c\Phi$. We say that an optimality criterion on F is *unit-invariant* if for every two families Φ and $\tilde{\Phi}$ and for every number $c > 0$ the following two conditions are true:

i) if Φ is better than $\tilde{\Phi}$ in the sense of this criterion (i.e., $\tilde{\Phi} < \Phi$), then $c\tilde{\Phi} < c\Phi$.

ii) if Φ is equivalent to $\tilde{\Phi}$ in the sense of this criterion (i.e., $\Phi \sim \tilde{\Phi}$), then $c\Phi \sim c\tilde{\Phi}$.

THEOREM. *If a family Φ is optimal in the sense of some optimality criterion that is final and unit-invariant, then every rescaling $f(u)$ from Φ is equivalent to $f(u) = log(u)$.*

(Proof is given in Section 5).

Comment. This means that the optimal rescalings are of the type $\gamma\, log(u) + \alpha$ for some real numbers $\gamma > 0$ and α.

4. CASE STUDY: APPLICATION OF LOGARITHMIC RESCALING TO FUZZY CONTROL (BRIEF DESCRIPTION)

Description of a plant. We design a control for chemical reaction within a constant volume, non-adiabatic, continuously stirred tank reactor (CSTR). The model that describes the CSTR is [M90]:

$$\dot{x}_1 = -x_1 + Da(1 - x_1)\, exp(x_2/(1 + x_2/\gamma))$$
$$\dot{x}_2 = -x_2 + BDa(1 - x_1)\, exp(x_2/(1 + x_2/\gamma)) - u(x_2 - x_c),$$

where x_1 is the conversion rate, x_2 is the dimensionless temperature, and u is the dimensionless heat transfer coefficient. The objective of the control is to stabilize the system (i.e., bring it closer to the equilibrium point).

What we did. We applied a logarithmic rescaling $x_2 \to X = log\, x_2$, and used membership functions with equal spacing for X. No further adjustment of membership functions was made.

Results. Even without any further adjustment the results of this control were comparable to the results of applying the intelligent "gain scheduled" (non-linear) PID controller ([HK85], [M90]). In other words, we got the control that was as good as the one generated by the state-of-art traditional control theory with respect to stability and controllability of the plant.

With respect to the computational complexity our fuzzy controller is much simpler.

Rescaling is necessary. Without the rescaling, we got a fuzzy control whose quality was much worse than that of a PID controller.

Details. The details of this case study were published in [VT92].

5. PROOF OF THE MAIN RESULT

The idea of this proof is as follows: first we prove that the optimal family is unit-invariant (in part 1), and from that, in part 2, we conclude that any function f from Φ satisfies a functional equation, whose solutions are known.

1. Let us first prove that the optimal family Φ_{opt} exists and is *unit-invariant* in the sense that $\Phi_{opt} = c\Phi_{opt}$ for all $c > 0$. Indeed, we assumed that the optimality criterion is final, therefore there exists a unique optimal family Φ_{opt}. Let's now prove that this optimal family is unit-invariant (this proof is practically the same as in [K90], [KQ91], or [KQF92]). The fact that Φ_{opt} is optimal means that for every other Φ, either $\Phi < \Phi_{opt}$ or $\Phi_{opt} \sim \Phi$. If $\Phi_{opt} \sim \Phi$ for some $\Phi \neq \Phi_{opt}$, then from the definition of the optimality criterion we can easily deduce that Φ is also optimal, which contradicts the fact that there is only one optimal family. So for every Φ either $\Phi < \Phi_{opt}$ or $\Phi_{opt} = \Phi$.

Take an arbitrary c and apply this conclusion to $\Phi = c\Phi_{opt}$. If $c\Phi_{opt} = \Phi < \Phi_{opt}$, then from the invariance of the optimality criterion (condition ii)) we conclude that $\Phi_{opt} < c^{-1}\Phi_{opt}$, and that conclusion contradicts the choice of Φ_{opt} as the optimal family. So $\Phi = c\Phi_{opt} < \Phi_{opt}$ is impossible, and therefore $\Phi_{opt} = \Phi$, i.e., $\Phi_{opt} = c\Phi_{opt}$, and the optimal family is really unit-invariant.

2. Let us now deduce the actual form of the functions $f(u)$ from the optimal family Φ_{opt}. If $f(u)$ is such a function, then the result $f(cu)$ of changing the unit of u to a c times smaller unit belongs to $c\Phi_{opt}$, and so, due to 1., it belongs to Φ_{opt}. But by the definition of a family all its functions can be obtained from each other by a linear transformation $Cf(u) + x_0$, therefore, $f(cu) = Cf(u) + x_0$ for some C and x_0. These values C and x_0 depend on c. So we arrive at the following functional equation for $f(u)$: $f(cu) = C(c)f(u) + x_0(c)$. In the survey on functional equations [A66] the solutions of this equation are not explicitly given, but a for a similar functional equation $f(x + y) = f(x)h(y) + k(y)$ all solutions are enumerated in Corollary 1 to Theorem 1, Section 3.1.2 of [A66]: they are $f(x) = \gamma x + \alpha$ and $f(x) = \gamma exp(cx) + \alpha$, where $\gamma \neq 0$, $c \neq 0$ and α are arbitrary constants. So, let us reduce our equation to the one with known solutions.

The only difference between these two equations is that we have a product, and we need a sum. There is a well known way to reduce product to a sum: turn to logarithms, because $log(ab) = log(a) + log(b)$. For simplicity let us use natural logatithms ln. So let us introduce new variables $X = ln(u)$ and $Y = ln(c)$. In terms of these new variables $x = exp(X)$, $c = exp(C)$. Substituting these values into our functional equation, and taking into consideration that $exp(X)exp(Y) = exp(X + Y)$, we conclude that $F(X + Y) = H(Y)F(X) + K(Y)$, where we denoted $F(X) = f(exp(X))$, $H(Y) = C(exp(Y))$, and $K(Y) = x_0(exp(Y))$. So according to the above-cited result, either $F(X) = \gamma X + \alpha$, or $F(X) = \gamma\, exp(cX) + \alpha$.

From $F(X) = f(exp(X))$, we conclude that $f(u) = F(ln(u))$, therefore either $f(u) = \gamma\, ln(u) + \alpha$, or $f(u) = \gamma\, exp(c\, ln(u)) + \alpha = \gamma u^c + \alpha$. In the second case the function $f(u)$ maps $(0, \infty)$ onto the interval (α, ∞), and we defined a rescaling as a function whose values run over all possible real numbers. So the second case is impossible, and $f(x) = \gamma\, ln(u) + \alpha$, which means that $f(u)$ is equivalent to a logarithm. Q.E.D.

6. CONCLUSIONS

One of the important steps in designing a fuzzy control is the choice of the membership functions for all the terms that the experts use. This choice strongly influences the quality of the resulting control.

For simple controlled systems, it is sufficient to have equally spaced membership functions, i.e., functions that have similar shape (usually triangular or trapezoid), and are located in intervals of equal length $..., [N - \Delta, N + \Delta], [N, N + 2\Delta], [N + \Delta, N + 3\Delta], ...$

For complicated systems this choice does not lead to a good fuzzy control, so it is necessary to tune the membership functions by applying neural networks or genetic algorithms. This is a very time-consuming procedure, and therefore, it is desirable to avoid it as much as possible.

We consider the case, when the equally spaced membership functions are inadequate because the control variable u can take only positive values. Such situations occur, for example, when we control the flux of the substances into a chemical reactor (e.g., the flux of fuel into an engine). Our idea is to "rescale" this variable, i.e., to use a new variable $u' = f(u)$, and to choose a function $f(u)$ in such a way that we can apply membership functions, that are equally spaced in u'.

We give a mathematical proof that the optimal rescaling is logarithmic ($f(u) = a\, log(u) + b$). We also show on a real-life example of a non-linear chemical reactor that the resulting fuzzy control,

without any further tuning of membership functions, can be comparable in quality with the best state-of-art non-linear controls of traditional control theory.

Acknowledgments. This research was supported by the Office of Naval Research grant No. N-00014-89-J-3123, NSF grant No. CDA-9015006, NASA Research grant NAG 9-482, and a grant from the Institute for Manufacturing and Materials Management. The authors are greatly thankful to all the participants of the 1992 International Fuzzy Systems and Intelligent Control Conference (Louisville, KY), especially to L. Zadeh and S. Smith, for valuable comments. One of the authors (V.K.) is also thankful to P. Suppes (Stanford) and L. Shepp (AT& T Bell Labs) for discussing the relevant mathematics.

REFERENCES

[A66] J. Aczel. *Lectures on functional equations and their applications.* Academic Press, NY–London, 1966.

[B91] H. R. Berenji. *Fuzzy logic controllers.* In: *An Introduction to Fuzzy Logic Applications in Intelligent Systems* (R. R. Yager, L. A. Zadeh. eds.), Kluwer Academic Publ., 1991.

[CZ72] S. S. L. Chang and L. A. Zadeh. *On fuzzy mapping and control*, IEEE Transactions on Systems, Man and Cybernetics, 1972, Vol. SMC-2, pp. 30–34.

[HK85] K. A. Hoo and J. C. Kantor. Chemical Engineering Communications, 1985, Vol. 37, No. 1.

[K90] V. Kreinovich. *Group-theoretic approach to intractable problems.* Lecture Notes in Computer Science, Springer, Berlin, Vol. 417, 1990, pp. 112–121.

[KK90] V. Kreinovich and S. Kumar. *Optimal choice of &- and ∨- operations for expert values* in: Proceedings of the 3rd University of New Brunswick Artificial Intelligence Workshop, Fredericton, N.B., Canada, 1990, pp. 169–178.

[KQ91] V. Kreinovich and C. Quintana. *Neural networks: what non-linearity to choose?*, Proceedings of the 4th University of New Brunswick Artificial Intelligence Workshop, Fredericton, N.B., Canada, 1991, pp. 627–637.

[KQF92] V. Kreinovich, C. Quintana, and O. Fuentes. *Genetic algorithms: what fitness scaling is optimal?.* University of Texas at El Paso, Computer Science Department, Technical Reports, January 1992.

[KQLFLKBR92] V. Kreinovich, C. Quintana, R. Lea, O. Fuentes, A. Lokshin, S. Kumar, I. Boricheva, and L. Reznik. *What non- linearity to choose? Mathematical foundations of fuzzy control.* Proceedings of the 1992 International Fuzzy Systems and Intelligent Control Conference, Louisville, KY, 1992, pp. 349–412.

[L90] C. C. Lee. *Fuzzy logic in control systems: fuzzy logic controller.* IEEE Transactions on Systems, Man and Cybernetics, 1990, Vol. 20, No. 2, pp. 404–435.

[M74] E. H. Mamdani. *Application of fuzzy algorithms for control of simple dynamic plant*, Proceedings of the IEE, 1974, Vol. 121, No. 12, pp. 1585–1588.

[M91] R. R. Mohler. *Nonlinear systems. Vol. I. Dynamics and control. Vol. II. Applications to bilinear control.* Prentice Hall, Englewood Cliffs, NJ, 1991.

[M90] N. D. Murray. *Nonlinear PID controllers*, Unpublished Master's Thesis, Virginia Polytechnic Institute and State University, September 1990.

[RSW92]. P. A. S. Ralston, K. E. Stoll, and T. L. Ward (editors). *Proceedings of the 1992 International Fuzzy Systems and Intelligent Control Conference*, Louisville, KY, 1992.

[S85] M. Sugeno (editor). *Industrial applications of fuzzy control*, North Holland, Amsterdam, 1985.

[SKLT71, 89] P. Suppes, D. H. Krant, R. D. Luce, and A. Tversky. *Foundations of measurement.* Academic Press, San Diego, CA. Vol. I, 1971; Vol. II, 1989, Vol. III, 1989.

[VT92] H. VanLangingham, A. Tsoukkas. *Application of fuzzy logic control to nonlinear process control.* Proceedings of the 1992 International Fuzzy Systems and Intelligent Control Conference, Louisville, KY, 1992, pp. 8–17.

[Z65] L. Zadeh. *Fuzzy sets.* Information and control, 1965, Vol. 8, pp. 338–353.

[Z71] L. A. Zadeh. *Towards a theory of fuzzy systems*, In: *Aspects of network and systems theory*, R. E. Kalman, N. DecLaris (eds.), Holt. Rinehart, Winston. 1971.

[Z73] L. A. Zadeh. *Outline of a new approach to the analysis of complex systems and decision processes*, IEEE Transactions on Systems, Man and Cybernetics, 1973, Vol. 3. pp. 28–44.

Genetic Learning in Rule-Based and Neural Systems

Robert E. Smith

Department of Engineering Mechanics
The University of Alabama
Box 870278
Tuscaloosa, Alabama 35487
phone: (205) 348-1618
fax: (205) 348-6419
rob@galab2.mh.ua.edu

The design of neural networks and fuzzy systems can involve complex, nonlinear, and ill-conditioned optimization problems. Often, traditional optimization schemes are inadequate or inapplicable for such tasks. Genetic Algorithms (GAs) are a class of optimization procedures whose mechanics are based on those of natural genetics. Mathematical arguments show how GAs bring substantial computational leverage to search problems, without requiring the mathematical characteristics often necessary for traditional optimization schemes (e.g. modality, continuity, availability of derivative information, etc.). GAs have proven effective in a variety of search tasks that arise in neural networks and fuzzy systems. This presentation begins by introducing the mechanism and theoretical underpinnings of GAs. GAs are then related to a class of rule-based machine learning systems called **learning classifier systems** (LCSs). An LCS implements a low-level production-system that uses a GA as its primary rule discovery mechanism. This presentation illustrates how, despite its rule-based framework, an LCS can be thought of as a competitive neural network. Neural network simulator code for an LCS is presented. In this context, the GA is doing more than optimizing an objective function. It is searching for an **ecology** of hidden nodes with limited connectivity. The GA attempts to evolve this ecology such that effective neural network performance results.

The GA is particularly well adapted to this task, given its naturally-inspired basis. The LCS/neural network analogy extends itself to other, more traditional neural networks. Conclusions to the presentation discuss the implications of using GAs in ecological search problems that arise in neural and fuzzy systems.

S24-63
182-21424
1002-4

Evolving Fuzzy Rules in a Learning Classifier System

Manuel Valenzuela-Rendon

mvalenzu@mtecv2.mty.itesm.mx
ITESM, Center for Artificial Intelligence
Sucursal de Correos "J" C.P. 64849
Monterrey, N.L., Mexico

The fuzzy classifier system (FCS) combines the ideas of fuzzy logic controllers (FLCs) and learning classifier systems (LCSs). It brings together the expressive powers of fuzzy logic as it has been applied in fuzzy controllers to express relations between continuous variables, and the ability of LCSs to evolve co-adapted sets of rules. The goal of the FCS is to develop a rule-based system capable of learning in a reinforcement regime, and that can potentially be used for process control.

Learning classifier systems are rule based machine-learning systems that can evolve rules in a reinforcement learning environment. In a LCS, automatic mechanisms adjust the strengths of rules according to their ability to receive payoff from the environment. A genetic algorithm runs over the population of rules, creating new rules by recombining those that have been successful in the past. The syntax commonly used in LCSs is designed for binary message matching, and therefore has great difficulty when dealing with continuous variables.

Fuzzy logic controllers have shown how fuzzy logic can be successfully applied to express in a few rules mappings between continuous variables. Their success is backed by a long list of applications. Nevertheless, in most of these applications, the designer of the FLC has developed the rules by hand, from interviews with the operator, from knowledge of the process, or from an operator's manual. No automatic way to develop sets of fuzzy rules for a FLC has gained recognition. In a FLC rules sets are stimulus-response, they do not take into account variables not directly supplied to the controller. The control of dynamic systems has usually been achieved by not only giving the reference and the error as inputs to the FLC, but also giving the derivative and integral of the error. Very complex processes might require the controller to take into account higher order derivatives or functions of variables that the designer might not be aware of. The FCS attempts to take advantage of LCSs ability to develop chains of rules automatically, and thus offer to the field of fuzzy control, characteristics not found in common FLCs.

Initial results show that the FCS can effectively create fuzzy rules that imitate the behavior of simple static systems. The current research work is directed towards increasing the learning rate of the FCS while retaining stability of that which has

been learned, and the imitation of more complex static systems. The next steps will be oriented towards the control of simple dynamic systems.

Adaptive Process Control Using Fuzzy Logic and Genetic Algorithms

By

C. L. Karr
U.S. Bureau of Mines, Tuscaloosa Research Center
Tuscaloosa, AL 35486-9777

ABSTRACT

Researchers at the U.S. Bureau of Mines have developed adaptive process control systems in which genetic algorithms (GAs) are used to augment fuzzy logic controllers (FLCs). GAs are search algorithms that rapidly locate near-optimum solutions to a wide spectrum of problems by modeling the search procedures of natural genetics. FLCs are rule based systems that efficiently manipulate a problem environment by modeling the "rule-of-thumb" strategy used in human decision making. Together, GAs and FLCs possess the capabilities necessary to produce powerful, efficient, and robust adaptive control systems. To perform efficiently, such control systems require a ***control element*** to manipulate the problem environment, an ***analysis element*** to recognize changes in the problem environment, and a ***learning element*** to adjust to the changes in the problem environment. Details of an overall adaptive control system are discussed. A specific laboratory acid-base pH system is used to demonstrate the ideas presented.

INTRODUCTION

The need for efficient process control has never been more important than it is today because of economic stresses forced on industry by processes of increased complexity and by intense competition in a world market. No industry is immune to the cost savings necessary to remain competitive; even traditional industries such as mineral processing (Kelly and Spottiswood, 1982), chemical engineering (Fogler, 1986), and wastewater treatment (Gottinger, 1991) have been forced to implement cost-cutting measures. Cost-cutting generally requires the implementation of emerging techniques that are often more complex than established procedures. The new processes that result are often characterized by rapidly changing process dynamics. Such systems prove difficult to control with conventional strategies, because these strategies lack an effective means of adapting to change. Furthermore, the mathematical tools employed for process control can be unduly complex even for simple systems.

In order to accommodate changing process dynamics yet avoid sluggish response times, adaptive control systems must alter their control strategies according to the current state of the process. Modern technology in the form of high-speed computers and artificial intelligence (AI) has opened the door for the development of control systems that adopt the approach to adaptive control used by humans, and perform more efficiently and with more flexibility than conventional control systems. Two powerful tools for adaptive control that have emerged from the field of AI are fuzzy logic (Zadeh, 1973) and genetic algorithms (GAs) (Goldberg, 1989).

The U.S. Bureau of Mines has developed an approach to the design of adaptive control systems, based on GAs and FLCs, that is effective in problem environments with rapidly changing dynamics. Additionally, the resulting controllers include a mechanism for handling inadequate feedback about the state or condition of the problem environment. Such controllers are more suitable than past control systems for recognizing, quantifying, and adapting to changes in the problem environment.

The adaptive control systems developed at the Bureau of Mines consist of a ***control element*** to manipulate the problem environment, an ***analysis element*** to recognize changes in the problem environment, and a ***learning element*** to adjust to the changes in the problem environment. Each component employs a GA, a FLC, or both, and each is described in this paper. A particular problem environment, a laboratory acid-base pH system,

serves as a forum for presenting the details of a Bureau-developed, adaptive controller. Preliminary results are presented to demonstrate the effectiveness of a GA-based FLC for each of the three individual elements. Details of the system will appear in a report by Karr and Gentry (1992).

PROBLEM ENVIRONMENT

In this section, a pH system is introduced to serve as a forum for presenting the details of a stand-alone, comprehensive, adaptive controller developed at the U.S. Bureau of Mines; emphasis is on the method not the application. The goal of the control system is to drive the pH to a setpoint. This is a non-trivial task since the pH system contains both nonlinearities and changing process dynamics. The nonlinearities occur because the output of pH sensors is proportional to the logarithm of hydrogen ion concentration. The source of the changing process dynamics will be described shortly.

A schematic of the pH system under consideration is shown in Fig. 1. The system consists of a beaker and five, valved input streams. The beaker initially contains a given volume of a solution having some known pH. The five, valved input streams into the beaker are divided into the two ***control input streams*** and the three ***external input streams***. Only the valves associated with the two control input streams can be adjusted by the controller. Additionally, as a constraint on the problem, these valves can only be adjusted a limited amount (0.5 mL/s/s, which is 20 pct of the maximum flow rate of 2.5 mL/s) to restrict pressure transients in the associated pumping systems.

The goal of the control problem is to drive the system pH to the desired setpoint in the shortest time possible by adjusting the valves on the two control input streams. Achieving this goal is made considerably more difficult by incorporating the potential for changing the process dynamics. These changing process dynamics come from three random changes that can be made to the pH system. First, the concentrations of the acid and base of the two control input streams can be changed randomly to be either 0.1 M HCl or 0.05 M HCl and 0.1 M NaOH or 0.05 M NaOH. Second, the valves on the external input streams can be randomly altered. This allows for the external addition of acid (0.05 M HCl), base (0.05 M CH_3COONa), and buffer (a combination of 0.1 M CH_3COOH and 0.1 M CH_3COONa) to the pH system. Note that the addition of a buffer is analogous to adding inertia to a mechanical system. Third, random changes are made to the setpoint to which the system pH is to be driven. These three random alterations in the system parameters dramatically alter the way in which the problem environment reacts to adjustments made by the controller to the valves on the control input streams. Furthermore, the controller receives no feedback concerning these random changes.

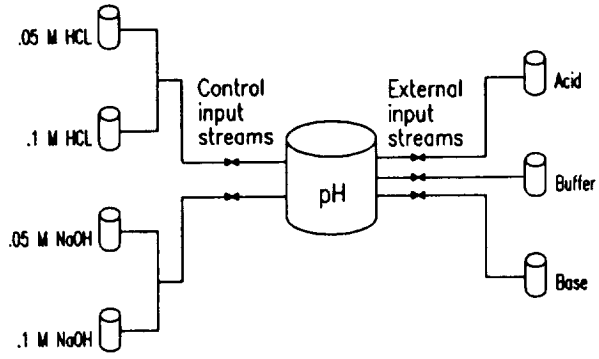


Fig. 1. Basic structure of the pH system.

The pH system was designed on a small scale so that experiments could be performed in limited laboratory space. Titrations were performed in a 1,000-mL beaker using a magnetic bar to stir the solution. Peristaltic pumps were used for the five input streams. An industrial pH electrode and transmitter sent signals through an analog-to-digital board to a 33-MHz 386 personal computer which implemented the control system.

STRUCTURE OF THE ADAPTIVE CONTROLLER

Figure 2 shows a schematic of the Bureau's adaptive control system. The heart of this control system is the loop consisting of the control element and the problem environment. The control element receives information from sensors in the problem environment concerning the status of the *condition variables*, i.e., pH and ΔpH. It then computes a desirable state for a set of *action variables*, i.e., flow rate of acid (Q_{ACID}) and flow rate of base (Q_{BASE}). These changes in the action variables force the problem environment toward the setpoint. This is the basic approach adopted for the design of virtually any closed loop control system, and in and of itself includes no mechanism for adaptive control.

The adaptive capabilities of the system shown in Fig. 2 are due to the analysis and learning elements. In general, the analysis element must recognize when a change in the problem environment has occurred. A "change," as it is used here, consists of any of the three random alterations to a parameter possible in the problem environment. (Of importance is the fact that all of these changes affect the response of the problem environment, otherwise it has no effect on the way in which the control element must act to efficiently manipulate the problem environment.) The analysis element uses information concerning the condition and action variables over some finite time period to recognize changes in the environment and to compute the new performance characteristics associated with these changes.

The new environment (the problem environment with the altered parameters) can pose many difficulties for the control element, because the control element is no longer manipulating the environment for which it was designed. Therefore, the algorithm that drives the control element must be altered. As shown in the schematic of Fig. 2, this task is accomplished by the learning element. The most efficient approach for the learning element to use to alter the control element is to utilize information concerning the past performance of the control system. The strategy used by the control, analysis, and learning elements of the stand-alone, comprehensive adaptive controller being developed by the U.S. Bureau of Mines is provided in the following sections.

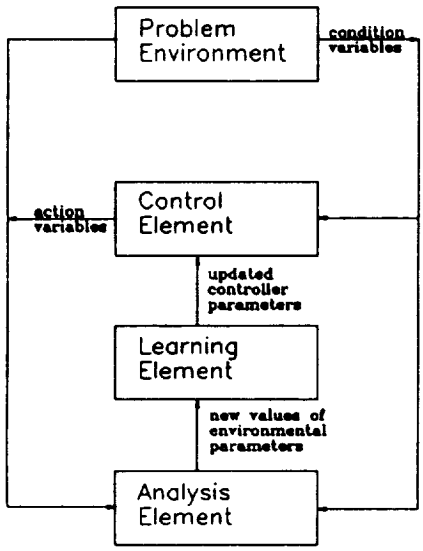


Fig. 2. Structure of the adaptive control system.

Control Element

The control element receives feedback from the pH system, and based on the current state of pH and ΔpH, must prescribe appropriate values of Q_{ACID} and Q_{BASE}. Any of a number of closed-loop controllers could be used for

this element. However, because of the flexibility needed in the control system as a whole, a FLC is employed. Like conventional rule-based systems (expert systems), FLCs use a set of production rules which are of the form:

IF {*condition*} THEN {*action*}

to arrive at appropriate control actions. The left-hand-side of the rules (the *condition* side) consists of combinations of the controlled variables (pH and ΔpH); the right-hand-side of the rules (the *action* side) consists of combinations of the manipulated variables (Q_{ACID} and Q_{BASE}). Unlike conventional expert systems, FLCs use rules that utilize fuzzy terms like those appearing in human rules-of-thumb. For example, a valid rule for a FLC used to manipulate the pH system is:

IF {ph is **VERY ACIDIC** and ΔpH is **SMALL**} THEN {Q_{BASE} is **LARGE** and Q_{ACID} is **ZERO**}.

This rule says that if the solution is very acidic and is not changing rapidly, the flow rate of the base should be made to be large and the flow rate of the acid should be made to be zero.

The fuzzy terms are subjective; they mean different things to different "experts," and can mean different things in varying situations. Fuzzy terms are assigned concrete meaning via fuzzy membership functions (Zadeh, 1973). The membership functions used in the control element to describe pH appear in Fig. 3. (As will be seen shortly, the learning element is capable of changing these membership functions in response to changes in the problem environment.) These membership functions are used in conjunction with the rule set to prescribe single, crisp values of the action variables (Q_{ACID} and Q_{BASE}). Unlike conventional expert systems, FLCs allow for the enactment of more than one rule at any given time. The single crisp action is computed using a weighted averaging technique that incorporates both a *min-max* operator and the *center-of-area* method (Karr, 1991). The following fuzzy terms were used, and therefore "defined" with membership functions, to describe the significant variables in the pH system:

pH	Very Acidic (**VA**), Acidic (**A**), Mildly Acidic (**MA**), Neutral (**N**), Mildly Basic (**MB**), Basic (**B**), and Very Basic (**VB**);
ΔpH	Small (**S**) and Large (**L**);
Q_{ACID}	Zero (**Z**), Very Small (**VS**),
Q_{BASE}	Small (**S**), Medium (**M**), and Large (**L**).

Although the pH system is quite complex, it is basically a titration system. An effective FLC for performing titrations can be written that contains only 14 rules. The 14 rules are necessary because there are seven fuzzy terms describing pH and two fuzzy terms describing ΔpH (7*2=14 rules to describe all possible combinations that could exist in the pH system as described by the fuzzy terms represented by the membership functions selected). Now, the rules selected for the control element are certainly inadequate to control the full-scale pH system; the one that includes the changing process dynamics. However, the performance of a FLC can be dramatically altered by changing the membership functions. This is equivalent to changing the definition of the terms used to describe the variables being considered by the controller. As will be seen shortly, GAs are powerful tools capable of rapidly locating efficient fuzzy membership functions that allow the controller to accommodate changes in the dynamics of the pH system.

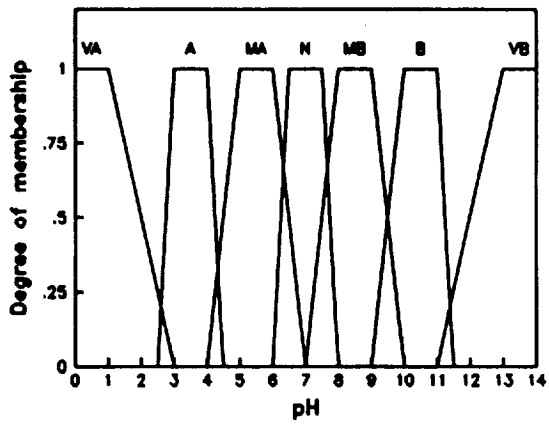


Fig. 3. pH membership functions.

Analysis Element

The analysis element recognizes changes in parameters associated with the problem environment not taken into account by the rules used in the control element. In the pH system, these parameters include: (1) the concentrations of the acid and base of the input control streams, (2) the flow rates of the acid, the base, and the buffer that are randomly altered, and (3) the system setpoint. Changes to any of these parameters can dramatically alter the way in which the system pH responds to additions of acid or base, thus forming a new problem environment requiring an altered control strategy. Recall that the FLC used for the control element presented includes none of these parameters in its 14 rules. Therefore, some mechanism for altering the prescribed actions must be included in the control system. But before the control element can be altered, the control system must recognize that the problem environment has changed, and compute the nature and magnitude of the changes.

The analysis element recognizes changes in the system parameters by comparing the response of the physical system to the response of a model of the pH system. In general, recognizing changes in the parameters associated with the problem environment requires the control system to store information concerning the past performance of the problem environment. This information is most effectively acquired through either a data base or a computer model. Storing such an extensive data base can be cumbersome and requires extensive computer memory. Fortunately, the dynamics of the pH system are well understood for buffered reactions, and can be modeled using a single cubic equation that can be solved for $[H_3O^+]$ ion concentrations, to directly yield the pH of the solution. In the approach adopted here, a computer model predicts the response of the laboratory pH system. This predicted response is compared to the response of the physical system. When the two responses differ by a threshold amount over a finite period of time, the physical pH system is considered to have been altered.

When the above approach is adopted, the problem of computing the new system parameters becomes a curve fitting problem (Karr, Stanley, and Scheiner, 1991). The parameters associated with the computer model produce a particular response to changes in the action variables. The parameters must be selected so that the response of the model matches the response of the actual problem environment.

An analysis element has been forged in which a GA is used to compute the values of the parameters associated with the pH system. When employing a GA in a search problem, there are basically two decisions that must be made: (1) how to code the parameters as bit strings and (2) how to evaluate the merit of each string (the fitness function must be defined). The GA used in the analysis element employs concatenated, mapped, unsigned binary coding (Karr and Gentry, 1992). The bit-strings produced by this coding strategy were of length 200:

the first 40 bits of the strings were used to represent the concentration of the acid on the control input stream, the second 40 bits were used to represent the concentration of the base on the control input stream, the third 40 bits were used to represent the flow rate of the acid of the external streams, and the final 80 bits were used to represent the flow rates of the buffer and the base of the external streams, respectively. The 40 bits associated with each individual parameter were read as a binary number, converted to decimal numbers (000 = 0, 001 = 1, 010 = 2, 011 = 3, etc.,), and mapped between minimum and maximum values according to the following:

$$C = C_{min} + \frac{b}{(2^m - 1)} (C_{max} - C_{min}) \tag{1}$$

where C is the value of the parameter in question, b is the binary value, m is the number of bits used to represent the particular parameter (40), and C_{min} and C_{max} are minimum and maximum values associated with each parameter that is being coded.

A fitness function has been employed that represents the quality of each bit-string; it provides a quantitative evaluation of how accurately the response of a model using the new model parameters matches the response of the actual physical system. The fitness function used in this application is:

$$f = \sum_{t=0s}^{t=100s} (pH_{model} - pH_{actual})^2. \tag{2}$$

With this definition of the fitness function, the problem becomes a minimization problem: the GA must minimize f, which as it has been defined, represents the difference between the response predicted by the model and the response of the laboratory system.

Figure 4 compares the response of the physical pH system to the response of the simulated pH system that uses the parameters determined by a GA. This figure shows that the responses of the computer model and the physical system are virtually identical, thereby demonstrating the effectiveness of a GA in this application. The GA was able to locate the correct parameters after only 500 function evaluations, where a function evaluation consisted of simulating the pH system for 100 seconds. Locating the correct parameters took approximately 20 seconds on a 386 personal computer. Industrial systems may mandate that a control action be taken in less than 20 seconds. In such cases, the time the GA is allotted to update the model parameters can be restricted. Once new parameters (and thus the new response characteristics of the problem environment) have been determined, the adaptive element must alter the control element.

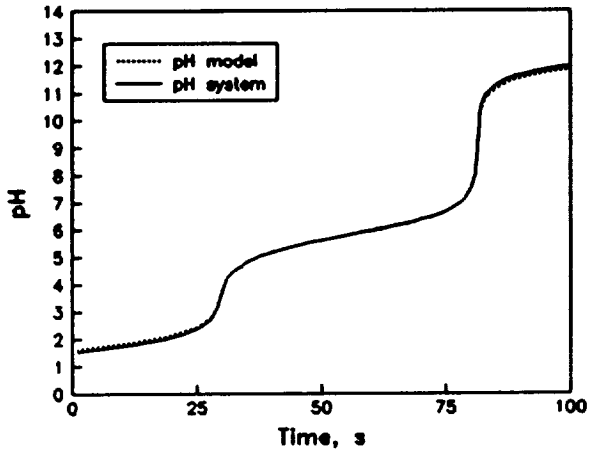


Fig. 4. Performance of an analysis element.

Learning Element

The learning element alters the control element in response to changes in the problem environment. It does so by altering the membership functions employed by the FLC of the control element. Since none of the randomly altered parameters appear in the FLC rule set, the only way to account for these conditions (outside of completely revamping the system) is to alter the membership functions employed by the FLC. These alterations consist of changing both the position and location of the trapezoids used to define the fuzzy terms.

Altering the membership functions (the definition of the fuzzy terms in the rule set) is consistent with the way humans control systems. Quite often, the rules-of-thumb humans use to manipulate a problem environment remain the same despite even dramatic changes to that environment; only the conditions under which the rules are applied are altered. This is basically the approach that is being taken when the fuzzy membership functions are altered.

The U.S. Bureau of Mines uses a GA to alter the membership functions associated with FLCs, and this technique has been well documented (Karr, 1991). A learning element that utilizes a GA to locate high-efficiency membership functions for the dynamic pH laboratory system has been designed and implemented.

The performance of a control system that uses a GA to alter the membership functions of its control element is demonstrated for two different situations. First, Fig. 5 compares the performance of the adaptive control system (one that changes its membership functions in response to changes in the system parameters) to a non-adaptive control system (one that ignores the changes in the system parameters). In this figure, the pH system has been perturbed by the addition of an acid (at 75 seconds), a base (at 125 seconds), and a buffer (at 175 seconds). In this case, the process dynamics are dramatically altered due to the addition of the buffer, and the adaptive controller is better.

Second, the concentrations of the acid and base the FLC uses to control pH are changed (those from the control input streams), which causes the system to respond differently. For example, if the 0.1 MHCl is the control input, the pH falls a certain amount when this acid is added. However, all other factors being the same, the pH will not fall as much when the same volume of the 0.05 M HCl is added. The results of this situation are summarized in Fig. 6. In this simulation, the concentration of the titrants is changed at 50 seconds. As above, the adaptive control system is more efficient.

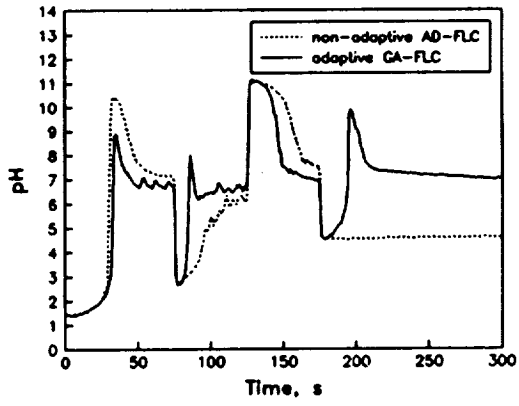


Fig. 5. External reagent additions.

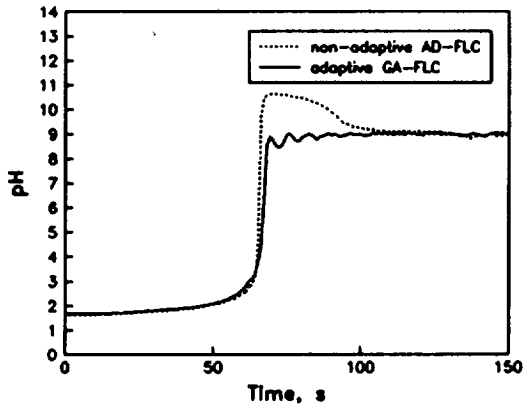


Fig. 6. Alteration of titrant concentrations.

SUMMARY

Scientists at the U.S. Bureau of Mines have developed an AI-based strategy for adaptive process control. This strategy uses GAs to fashion three components necessary for a robust, comprehensive adaptive process control system: (1) a control element to manipulate the problem environment, (2) an analysis element to recognize changes in the problem environment, and (3) a learning element to adjust to changes in the problem environment. The application of this strategy to a laboratory pH system has been described.

REFERENCES

Fogler, H. S. (1986). *Elements of Chemical Reaction Engineering*. Prentice-Hall, Englewood Cliffs, NJ.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading, MA.

Gottinger, W. W. (1991). *Economic Models and Applications of Solid Waste Management*. Gordon and Breach Science Publishers, New York, NY.

Karr, C. L. (1991). Genetic algorithms for fuzzy logic controllers. *AI Expert*, 6, 26-33.

Karr, C. L. and Gentry, E. J. (1992, in press). *An Adaptive System for Process Control*. U.S. Bureau of Mines Report of Investigations.

Karr, C. L., Stanley, D. A., and Scheiner, B. J. (1991). *A Genetic Algorithm Applied to Least Squares Curve Fitting*. U.S. Bureau of Mines Report of Investigations No. 9339.

Kelly, E. G. and Spottiswood, D. J. (1982). *Introduction to Mineral Processing*. John Wiley & Sons, New York, NY.

Zadeh, L. A. (1973). Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-3, 28-44.

S26-63
15464

N93-22377

Design of Fuzzy System by NNs and Realization of Adaptability

Hideyuki TAKAGI
Computer Science Division
University of California at Berkeley
(Central Research Laboratories, Matsushita Electric Industrial Co., Ltd.)

1. Proposal of NN-driven Fuzzy Reasoning (1988)

The issue of designing and tuning fuzzy membership functions by neural networks (NNs) was started by NN-driven Fuzzy Reasoning in 1988. NN-driven Fuzzy Reasoning involves a NN embedded in the fuzzy system which generates membership values. In conventional fuzzy system design, the membership function are hand-crafted by trial and error for each input variable. In contrast, NN-driven Fuzzy Reasoning considers several variables simultaneously and can design a multidimensional, nonlinear membership function for the entire subspace.

2. Knowledge/Skill acquisition by NN-driven Fuzzy Reasoning. (1989)

Consider the problem of balancing a pole starting with an initial swing from the hanging-down position. NN-driven Fuzzy Reasoning can process the raw data generated by a human adept at this task and can learn to infer the rules necessary for executing this task. This method has shown its ability to acquire knowledge and skill which is difficult to convey using language but is easily demonstrated.

3. Simplified design method for membership functions (1990).

Two issues affected by NN-driven Fuzzy Reasoning emerged in 1990. One was the design of structured NNs (Neural networks designed on Approximate Reasoning Architecture). The other concerned shortening the design time of membership functions so that the techniques could be used in a practical setting.

This simplified method works with one-dimensional, triangular membership functions instead of the fully general, multidimensional, nonlinear shapes, but this restriction helps speed up the design phase significantly. Currently this method is used for the design of several consumer products involving fuzzy logic (FL) and NNs by Matsushita Electric group.

4. Application of NN and FL in consumer products (1991).

Following the application of such technology in an air-conditioner in 1990, several consumer products using FL and NNs have appeared on the market in

1991. Till autumn 1991, fourteen such products had appeared on the Japanese market. In the context of consumer products, NNs have been put to use in the following five ways : (I) development tools, (2) independently of the fuzzy system, (3) as a correcting mechanism, (4) in cascade combination with FL, and (5) for learning user preferences. Equipment designed with the method mentioned in Sec. 3 falls in category (1).

5. Realization of Adaptability : Current Issues

Achieving adaptability is an important concern when fusing NNs and FL. It is too inflexible to pre-program things that depend on the user's preferences or environment. What is needed is some way to learn the usage patterns and adjust the rules using the adaptive capability of NNs. Category (5) in the previous section is intended to follow this direction.

Realization of "equipment of which handling easiness is improved as it is used more" corresponds to incremental learning in NNs. Suppose we wish to modify the equipment based on data provide by the user's actions and environment. In this case, the additional learning should have the following properties: (a) do not use all of the past training data, (b) the changes should have local effect only, in some sense, (c) training data which is more recent and supersedes older data should be recognized as such and the older information forgotten, (d) if the changes lead to violation of strict safety constraints, such data is potentially harmful and should be ignored.

6. Realization of Adaptability : Proposed Algorithm to Extract Boundary of Datasets.

Partitioning the input space is essential for determining the rulebase, such as in a fuzzy controller. Adaptive rule modification corresponds to modifying the partitioned subsets of the space. If the new data is on the boundary of the distribution of the training set, then the problem can be solved so that the four requirements in Sec. 5 above are obeyed.

This algorithm for extracting boundary data uses n-dimensional ellipses of which all axes but the major axis are equal. These shapes are used to eliminate data which lies inside the boundary, leaving the boundary points of the training dataset.

If new data is introduced on top of a boundary as shown in Fig. 1 (a), the algorithm will modify the old boundary and incorporate the new data as shown in Fig. 1 (b). This is a modification of the rule partitioning.

N93-22378

Improvement on Fuzzy Controller Design Techniques

Professor Paul P. Wang
Department of Electrical Engineering
Duke University
Durham, NC 27706

This paper addresses three main issues, which are somewhat interrelated.

The first issue deals with the classification or types of fuzzy controllers. Careful examination of the fuzzy controllers designed by various engineers reveals distinctive classes of fuzzy controllers. Classification is believed to be helpful from different perspectives.

The second issue deals with the design according to specifications. experiments related to the tuning of fuzzy controllers, according to the specification, will be discussed. General design procedure, hopefully, can be outlined in order to ease the burden of a design engineer.

The third issue deals with the simplicity an limitation of the rule-based IF-THEN logical statements. The methodology of fuzzy-constraint network is proposed here as an alternative to the design practice at present. It is our belief that predicate calculus and the first order logic possess much more expressive power.

Throughout the talk, the integration of the fuzzy control technology with the conventional control system design techniques will be our focus.

AUTHOR INDEX

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	January 1993	Conference Publication

4. TITLE AND SUBTITLE
Proceedings of the Third International Workshop on Neural Networks and Fuzzy Logic

5. FUNDING NUMBERS

6. AUTHOR(S)
Christopher J. Culbert, Editor

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
NASA Lyndon B. Johnson Space Center
Information Technology Division
Houston, TX 77058

8. PERFORMING ORGANIZATION REPORT NUMBER
NASA CP-10111
Vol. I Vol. II

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Washington, D.C. 20546

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
S-701

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Unclassified/Unlimited
Subject category 63
Volume I - unlimited
Volume II - unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Documented here are papers presented at the Neural Networks and Fuzzy Logic Workshop sponsored by the National Aeronautics and Space Administration and cosponsored by the University of Houston, Clear Lake. The workshop was held June 1-3, 1992 at the Lyndon B. Johnson Space Center in Houston, Texas. During the three days approximately 50 papers were presented. Technical topics addressed included adaptive system; learning algorithms; network architectures; vision; robotics; neurobiological connections; speech recognition and synthesis; fuzzy set theory and application, control, and dynamics processing; space applications; fuzzy logic and neural network computers; approximate reasoning; and multiobject decision making.

14. SUBJECT TERMS
fuzzy logic; non-Lipschitzian dynamics, parallel distributed models; algorithms; neural network; spatiotemporal patterns; neuron ring; fuzzy controllers; signal processing; pattern recognition

15. NUMBER OF PAGES
393

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	unlimited